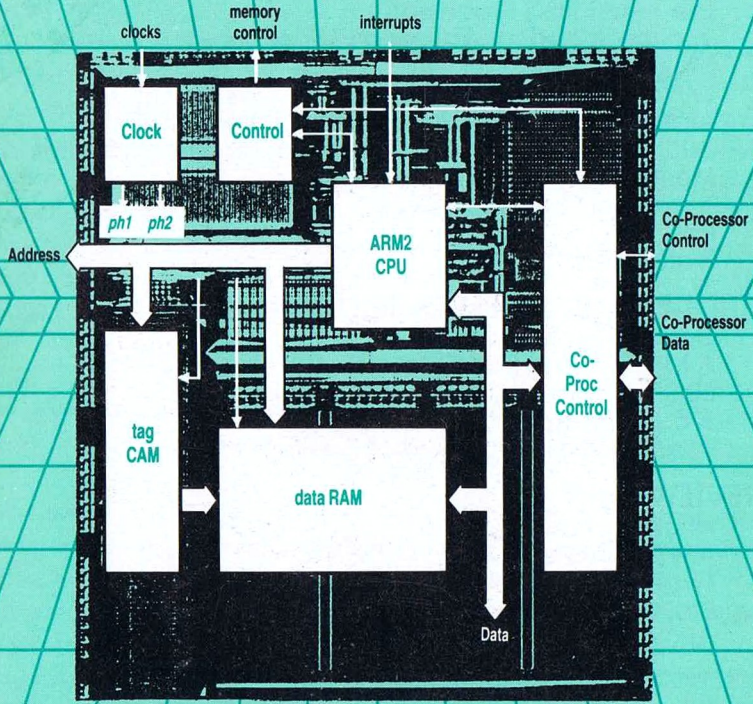
Volume 2
Issue 8

July/
August
1989

Price £1.20

RISC USER



The ARM3 RISC Processor

THE MAGAZINE AND SUPPORT GROUP EXCLUSIVELY FOR USERS OF THE ARCHIMEDES

Volume 2 Issue 8
July/August 1989

RISC USER

CONTENTS

FEATURES

News	4
The ARM3 RISC Processor	7
Into the ARC: Procedures and Functions	19
Making the Most of 1st Word Plus	31
Win an A3000 Colour System - Competition	35
Archimedes Visuals: Special effects with VIDC	45
ARC Procedure Library: General purpose input function	49
Memory Allocation Revisited	59
Technical Queries	61
Postbag	63
Hints & Tips	65

UTILITIES AND APPLICATIONS

Desktop Basic Handler	11
A Multi-Tasking Disc Formatter	23
BBC to Archimedes Screen Converter	39
A Basic Program Cross-Referencer	43
Basic Squeeze	51

REVIEWS

Ancestry from Minerva	16
ArcComm Review	28
Two Basic Compilers	36
Brainsoft's Multiple I/O Podule	40
ARCade	57

RISC User is published by BEEBUG Ltd.

Co-Editors: Mike Williams, Dr Lee Calcraft
Assistant Editor: Kristina Lucas
Technical Editor: David Spencer
Technical Assistant: Glynn Clements
Production assistant: Sheila Stoneman
Advertising: Sarah Shrive
Subscriptions: Mandy Mileham

BEEBUG Ltd (c) 1989

All rights reserved. No part of this publication may be reproduced without prior written permission of the Publisher. The Publisher cannot accept any responsibility whatsoever for errors in articles, programs, or advertisements published. The opinions expressed on the pages of this journal are those of the authors and do not necessarily represent those of the Publisher, BEEBUG Limited.

BEEBUG, Dolphin Place, Holywell Hill, St.Albans,
Herts AL1 1EX. Tel. St.Albans (0727) 40303

Printed by Arlon House

The Archimedes Magazine
and Support Group.

EDITORIAL

THE ARM 3

As you will have seen in last month's RISC User, Aleph One Ltd. are developing an ARM 3 upgrade board for the Arc. Acorn's ARM 3 is a development of the ARM 2 RISC chip used in all of the current Archimedes range (including the A3000 and the A400/1 series). It contains a 4K on-board memory cache, and promises very significant speed increases over the ARM 2. At the rumoured price of around £600 for the upgrade, it will be cheaper than Acorn's long-awaited floating point co-processor board (this is expected to cost around £700), and for many applications will offer a much greater speed increase. The floating point co-processor, for example will not make Basic run any faster at all. Because of the importance of the ARM 3, this month's RISC User features an article on the chip by Acorn's Roger Wilson, the author of Basic V.

RISC USER

Thanks to all who have answered last month's questionnaire. We have been very pleased with the enthusiastic response, and we will publish the results in a future issue of the magazine.

You will also see that one of the items in this month's magazine features a program which is too long to be listed in the magazine itself, but which is available on the magazine disc. While we do not intend to treat more than one program per issue in this way, nor to devote more than a page to its description, this approach does allow us to supply programs which would otherwise not see the light of day because of their length.

Finally, please note that as from early August, our new address will be:

117 Hatfield Road,
St Albans,
Herts AL1 4JS.

For full details, see the insert with this issue.

See you at the BBC Acorn User Show (July 21-23), or at our forthcoming Open Day in St Albans.

This month's telesoftware password is ***cranberry.***

DTP IS HERE

Spring seems to be the time for desktop publishing packages. Firstly, Acorn's long awaited system is now available. It is unimaginatively called *Desktop Publisher,* and is in fact a modified version of the Timeworks package for the Atari ST. The package includes the ability to use a wide range of text sizes and styles, and also allows pictures to be incorporated. *Desktop Publisher* costs £171.35 (inc. VAT) and will be available from all Acorn dealers.

Another DTP package is *Pixel Perfect* from AVP, which was originally launched some time ago, but has now been rewritten to take advantage of RISC OS. Unlike Acorn's system, *Pixel Perfect* is aimed mainly at the educational market. More details can be obtained from AVP Computing, School Hill Centre, Chepstow, Gwent NP6 5PH, tel. (0291) 625439.

Meanwhile BEEBUG has developed its own DTP package, and details can be found on the following page, and in the current retail catalogue.

MACH PRODUCTS

MACH Technology, a new company on the Archimedes scene, promises a wide range of quality, low cost products for the Archimedes. Among their first offerings are a 20Mb hard disc upgrade for the 300 series at £249 (all prices inc. VAT), and an 80286 based PC card for £300. This latter device will allow PC software to run many times faster than it can be under the PC emulator. Also promised is a hardware floating point co-processor which will rival Acorn's own unit, which is due for release very soon. On the software front, Mach are offering a full desktop publishing package for £89.95, a Basic compiler for £39.95, Hyperbase, a graphical database, for £49.95, and a Modula 2 compiler for £89.95. More details, and information on availability, can be obtained from Mach Technology, 75 Hydewood Road, Little Yeldham, Essex CO9 4QX, tel. (0787) 238125.

THE PIPEDREAM GROWS

Colton Software has totally rewritten their popular integrated word processor *Pipedream* to make it fully RISC OS compatible. *Pipedream 3* is fully multi-tasking and has a number of new features, including the ability to import View and Viewsheet files directly. New spreadsheet features allow for multiple sheets with cross references, together with several new functions. There is a macro recording facility to enable sequences of actions to be remembered, and files can be exported directly into Acorn's *Desktop Publisher. Pipedream 3* includes an in-built spelling checker, and costs £169.05 (inc. VAT). The package will be on sale at the BBC Acorn User Show in July.

SCSI HARD DISCS

Oak Computers, best know for the Oak PDT design package, have developed a Small Computer Systems Interface (SCSI) podule for the Archimedes. SCSI is an intelligent interface designed mainly for connecting mass-storage peripherals to computers. The Oak unit will allow up to four hard disc drives to be connected, each with a capacity of up to 512Mb. One use of the new board will be for owners of model B hard discs, who wish to connect them to the Archimedes. Price and availability are not yet known, though Oak say that the interface will be on display at the BBC Acorn User Show in July.

I SPY A BIRD

Birdlog is a public domain database specially tailored to the needs of bird watchers. The package allows you to keep track of the comings and goings of different species, in a number of places, on a day-by-day basis. Facilities exist for statistical analysis of the data, which can subsequently be exported to a word processor. *Birdlog* is supplied with a 9600-word user guide on disc, and can be obtained by sending £5.99 (to cover disc and costs) to David Pilling, PO Box 22, Thornton Cleveleys, Blackpool, FY5 1LR.

RISC OS SUPPLY

RISC User members who feel that they have been waiting for ages for RISC OS upgrades will be pleased to know that Acorn believe the supply problem to be over. By the time you read this, all outstanding orders from BEEBUG should have been satisfied, and RISC OS will be available from stock. Acorn say that the problem has been equally bad for all dealers, and is a result of the time taken by Japanese companies to produce the required quantities of ROM chips.

BEEBUG NEWS

On this page we take a look at some of the exciting new products and developments at BEEBUG which will shortly be available to RISC User members.

NEW ARCHIMEDES PRODUCTS

BEEBUG has been busy developing a number of new products for the Archimedes. First and foremost is a full-feature **desktop publishing** and **word processing** package. It is fully WIMP based and makes extensive use of 'fancy fonts'. As well as entering and laying out text in many styles and sizes, sprites and draw files can be included in the document. There is also a fast built-in spelling checker.

A perfect companion to this is our new 200 dot-per-inch handheld **scanner** and **interface podule.** This allows photographs and pages of text to be read in as sprites, and then incorporated into documents or used in other packages. The scanner will be available in A6 and A4 widths, with an optional sheet-feeder for the A4 model.

The BEEBUG **Hard Disc Backup** will be of particular interest to all hard disc users. This WIMP driven package allows either full or incremental backup of files, and uses a compression technique to reduce the number of floppies needed. Also included is a multi-file archiver and de-archiver, and a utility to locate files on your hard disc.

All of these products can be seen at the BBC Acorn User Show (see below), and more details are given in the current retail catalogue.

BEEBUG EXPANDS PREMISES AND PRODUCT RANGE

BEEBUG and RISC User are moving to new and larger premises in St. Albans in early August. This long awaited move will enable us to offer a number of new and improved services.

The new showroom is about double the size of the existing one, enabling us to have more equipment on display. It will also allow us to offer Aries and Amstrad PC's for customers who have requirements in this area. We have now employed a full-time engineer and can offer a faster turnaround on repairs. Additionally, in response to requests from members, we now have a training facility, and will shortly be offering competitively priced courses on a variety of subjects.

Please see the enclosed leaflet for more details of our new premises, and an invitation to our open day to see our new facilities and meet the staff.

ORDER AND SUBSCRIPTION PROCESSING

We have recently installed a major networked computer system to handle order processing and subscriptions. This means that whether you phone in with an order or subscription, we will have all your details to hand. Our telesales staff, Julie and Debbie, will key your order straight into the computer, which now enables us to despatch over 80% of orders on the day they are received. On subscription processing, Mandy and Sarah can similarly renew your subscription or change your address etc., instantly over the phone.

The new system also keeps control of our large stock of hardware and software, thereby ensuring that we can re-order items before running out. As a consequence of this new system, your personal membership number is now more important than ever, and should always be quoted over the phone, or in correspondence.

BBC ACORN USER SHOW

The BBC Acorn User show at Alexandra Palace from 21st - 23rd July is the only Acorn-specific show this year. BEEBUG and RISC User will be there on **stand 55**. We will be demonstrating all the new products mentioned above, along with many other items including the new **Acorn A3000** computer (currently on demonstration in our showroom). The magazine and technical staff will be on hand to chat with members and try and help with any problems. Alternatively, why not take the opportunity to meet the faces behind the names seen on the pages of BEEBUG and RISC User.

NEW CATALOGUE

With this issue of RISC User you will have received a much larger retail catalogue than normal. This new format catalogue allows us to give more details, and pictures of products. Included in this new catalogue is the full range of Amstrad and Aries PCs which we stock. RU

BBC Acorn User Show

BBC ACORN USER, with the backing of the BBC and the enthusiastic support of Acorn Computers, have together planned what could be the most exciting computing event of the year.

The new, all-action BBC ACORN USER SHOW is back, at the first home of BBC TV – Alexandra Palace featuring:-

- new product launches
- new software
- massive computing exhibition
- informative seminars
- technical clinics
- workshops
- demonstrations

All this and more will make the new BBC ACORN USER SHOW a real must for everyone interested in computers and their applications.

There are daily competitions, free draws with fabulous prizes to be won and celebrity guest appearances.

THE 1989 BBC ACORN USER show

Facilities at the tastefully restored Alexandra Palace are simply superb and it's so easy to get to by road, rail or tube. If you're driving you'll be glad to know there are 2000 free car parking spaces. So why not make a day of it? Apart from the beautiful grounds you'll find something interesting and exciting around every corner.

Why not save time and money by booking your ticket in advance? Cut the coupon now and ensure your priority booking.

ALEXANDRA PALACE · LONDON
Friday July 21st 3pm-9pm
Saturday July 22nd 10am-6pm
Sunday July 23rd 10am-6pm

BOOK NOW AND SAVE

Advance Tickets: Adults £2.50 Under 16s £1.50

At the door: Adults £3.50 Under 16s £2.50

Please send me _____ Adult Tickets at £2.50 each

Please send me _____ Under 16s Tickets at £1.50 each.

I enclose cheque/PO for £_____ payable to SAFESELL Ltd., Market House, Cross Road, Tadworth, Surrey KT20 5SR.

DON'T SEND CASH.

Name _____

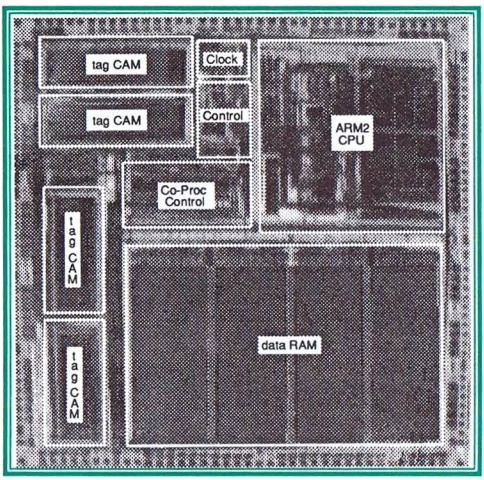
Address _____

RU1

The ARM3 RISC Processor

Roger Wilson, Manager of Software Development (AR&D) at Acorn, lifts the lid off Acorn's latest RISC Processor

The current range of ARM based computers (Archimedes A410/1, A420/1, A440/1, R140 and BBC A3000) all use the Acorn-designed ARM2 VLSI chip, a small, cheap and fast RISC microprocessor, running at a clock speed of 8MHz. In the laboratory ARM2 has run at 20MHz by using very expensive static RAM, but the requirements of the commercial market limit the current machines to running ARM2 at 4 or 8MHz with ordinary memory systems built from 120nS dynamic RAM.



ARM3 floorplan

ARM3 is designed to run at higher speeds while using standard memory. It can do this because of the typical behaviour of a program: instructions and data are re-used many times (for example in loops) in order to complete a task. By copying instructions and data into high speed memory as they are first encountered, the later references will run faster. The high speed memory where these copies are placed is called a *cache* (pronounced "cash").

ARM3 contains within it an enhanced ARM2 processor plus 4Kbytes of cache memory built onto the same silicon chip. This may not seem very much compared to the 4Mbytes of memory inside the A440, but it is enough to provide performance benefits on all programs because of the generally small nature of program loops. Being smaller than the system's main memory, the cache memory has, in fact, to remember two things: the instruction or data itself, and which address it came from: this latter piece of information is referred to as the *tag* just like the address tag (label) on a parcel.

When the processor requires information, the tags are checked to see if the information is in the cache: if it is, the associated data is provided, otherwise an external memory cycle is used. To reduce the size of the structures used by the tags, each tag guards four 32-bit words of data (so 256 tags are needed for the 4K cache on ARM3). The ARM3 thus reads four words into the cache even if the programmer specified just one byte. This is actually another advantage: the adjacent locations (which will probably be referred to subsequently in any case) have already been copied into the cache by the time the program gets round to using them. And four-word transfers are advantageous with the MEMC memory controller - reading the three extra values takes a little over twice as long. Each data write is checked against the entries in the cache, and if a match is found, the information in the cache is updated.

An interesting part of the design is the way the position of a new entry in the cache is selected. As new information is copied from main memory to the cache, something currently in the cache must be overwritten with the new data. For ARM3 this is selected on a purely random basis! Simulations have shown that more intelligent algorithms (such as discarding the least recently used entry) do no better than a random one, and would sometimes do much worse.

The cache is caching virtually addressed data (since the translation of virtual addresses to physical addresses occurs inside the MEMC chip) and so has to have an additional state to

deal with information that the supervisor mode can read but the user mode can't. It must also be cleared when the operating system, whether RISC OS or RISC iX (on the R140 UNIX machine) addresses a new task. And it must avoid placing values which change by themselves (e.g. in the input/output system) in the cache.

For efficiency it is also a good idea for the chip to be capable of not caching various things: for example the screen memory is a poor subject for caching, being written first and read second (if at all), and being decidedly larger than the cache. And there are areas readable at two virtual addresses (like the screen and the physically addressed RAM), and even places where reads and writes are to different devices (e.g. read the ROM and write the MEMC control registers).

To support all of these miscellaneous facilities there are some cache control registers in ARM3 (which only the supervisor can change). These three registers split the memory into 2Mbyte sections providing 3 options:

1. Uncachable: set I/O areas, doubly mapped DRAM etc. to uncachable
2. Updatable: writes are copied to the cache
3. Disruptive: writes will invalidate the cache

By setting these control bits appropriately, RISC OS and RISC iX systems can be run totally compatibly on the ARM3 with no changes to the applications or the operating systems. When reset, ARM3 disables its cache, thus allowing these operating systems to be started as if ARM3 were an ARM2, a cache initialisation program can then be run to turn the cache on.

In addition to "simply" going faster, ARM3 provides some new facilities. The first is to do with the co-processor bus: if all the accesses to memory are fulfilled by the cache, how does the co-processor (currently attached to the data bus between ARM2 and MEMC) function? ARM3 has a separate co-processor bus to which existing co-processors can be attached.

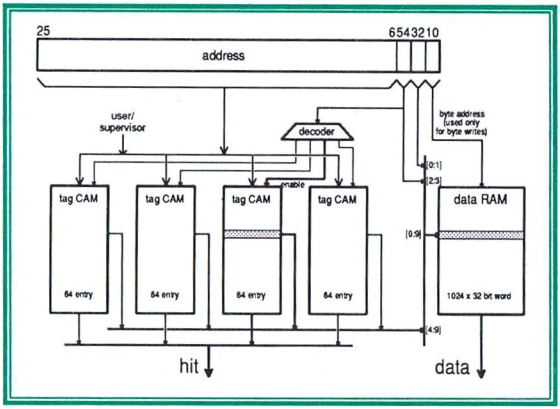
The second facility is more interesting: a single ARM3 uses only 1/4 to 1/2 of the data bus bandwidth provided by MEMC. If the video system isn't using it, why not add another ARM3? (though its not as simple as it sounds!) If one is to use multiple ARM3s they need a secure method of communication which will return a valid answer even if all the processors try to alter the same location simultaneously. Ordinary LDR/STR communication is not secure: if one processor moves a value to a register, adds one to it and writes it back, another processor might have written information to the same location in the interim. ARM3 has a LOCK signal added to its bus interface and an instruction which uses it: SWP swaps a byte or word between register and memory, doing the load and store with LOCK asserted so that other processors will not gain access to the bus in between these operations. This enables a particular processor to set a flag (and be sure that the other processor has not simultaneously done the same thing) as an indication that a certain area of memory should not be accessed by the other processor.

Process	1.5um DLM CMOS
Chip size	8.72mm x 9.95mm
No. Transistors (RAM/CAM)	309,656 (206,454 / 62,973)
Cache size	4Kbytes
Cache clock speed	25MHz
Performance (120ns DRAM main memory)	25 MIPS peak 12 MIPS sustained
Power consumption	1 Watt
Package	160 pin Quad Flat Pack
No. Power pins	41

ARM3 characteristics

Obviously, with all these extra signals (and others like the high speed clock which runs the processor core), ARM3 is not pin-compatible with ARM2! It has 160 pins (including 41 connected to the power lines) and is in a quad flat package with 40 leads a side. ARM3 is, however, signal compatible with the existing chip set so that it can use MEMC, VIDC, IOC and the FPPC (floating point protocol converter) in new designs. ARM3 has been tested on 1.5 micron double level metal CMOS process where its 309,656 transistors use an area 8.72 by 9.95 mm. Chips run between 20 and 30MHz dissipating up to one Watt of

power. However, it will be reduced to a 1 micron process before high volume production, and this reduction in size should allow an increase in maximum clock rate.



ARM3 cache block diagram

So how fast does it go? It's difficult to say since it does depend on so many things: screen mode, ARM3 clock speed, program, operating system.... For a 20MHz ARM3 running the Dhrystone 2.1 program under RISC OS in mode 0, the speed is increased by a factor of 3. But in higher screen modes like mode 20, the ARM3 hardly slows down at all, while the ARM2 runs at only 50-60% of its original speed, so the relative improvement is even greater. And if you spend a lot of time in mode 21..... Conversely a simple mode like mode 18 and a different program - Smalltalk Express' Smalltalk-80 - gives only a 1.3 - 1.5 times improvement. In a graphical environment like RISC OS's Desktop, type faces, scrolling, panning, sprites all cache very well and result in a distinctly "light feeling" to all the objects: they move around very easily and quickly.

Although all programs running under RISC OS or RISC iX will run on ARM3, there are some things that can be done to optimise the performance on ARM3, usually without harming the performance on ARM2:

1. Don't unroll loops too many times. A common technique to speed up execution of loops is to "unroll" them, replicating the body of the loop several times and just having one end of loop statement. On an uncached processor, this will clearly speed things up since fewer statements get executed. On a cached processor the algorithm fills more cache space than necessary (thus discarding other useful information) and re-uses the items fewer times.

2. Don't use STR, STRB. On the cached processor, store instructions always halt the processor while they go out to memory. The normal three cycle cost of a store is, therefore, greatly magnified, becoming 6 or more cycles for a 20MHz processor, because of the relatively slow speed of off-chip RAM access. Collect store instructions together, replacing STRBs with STR, and STRs with STM. Load instructions also suffer, but at least they have the chance of being in the cache. I recently re-wrote the store floating point algorithm in BBC BASIC, which has to store 5 bytes in consecutive locations, to make it ARM3 friendly. Before the change it took 20 stores, 0 loads and 12 data operations to do four of them. Afterwards it took 10 stores, 2 loads and 32 data operations. The changed algorithm is slightly faster for an ARM2 and much faster on an ARM3.

3. Try to increase the locality of the code. Group the code by use: put together all the bits which operate at the same time. Use common sub-routines to concentrate critical sections into the cache.

Many similar techniques will occur to the proficient programmer!

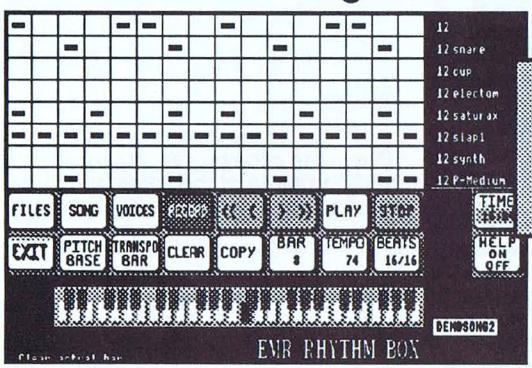
Since ARM3 is relatively new there are only a few places where you can find out more. VLSI Technology Inc (Milton Keynes) will be selling the chip as VL86C020 and will have data manuals etc. Acorn and VLSI Technology Inc have provided papers to various conferences this year describing the implementation of the chip. A book describing RISC generally and ARM2 and ARM3 in particular *VLSI RISC Architecture and Organisation* by S. Furber; published by Marcel Dekker, Inc, ISBN 0-8247-8151-1. Price $119.50, unfortunately, but some of you must be able to buttonhole a friendly librarian! RU

I've got RHYTHM Box -have you?

It's so easy - just COUNT it through:

1 Collect your SOUNDS within the ARC
  - great samples FREE in Rhythm Box!
2 Choose eight new sounds for ANY bar
3 Click on a box to HEAR a sound
4 Play the keys and set the PITCH
5 SYNCOPATE live or step by step
6 Do the MIX and get it dynamic
7 Arrange the bars and make the SONG
8 Connect the stereo and play a HIT!

There's nothing like **RHYTHM BOX** -try it and see!



ONLY £29.95

Inc VAT. Add £2.50 P&P.
All EMR programs are Risc OS Compatible.

RHYTHM BOX is part of EMR's ARPEGGIO MUSIC SYSTEM for the ARCHIMEDES computers

Tel: 0702 335747 Fax: 0702 431076
Access & Visa by phone. Send for info.

Archimedes is a musician - ask EMR!

Electromusic Research (EMR) Ltd
14 Mount Close, Wickford, Essex SS11 8HG

DESKTOP BASIC HANDLER

Simplify the development of Basic programs from within the RISC OS Desktop using this utility by David Spencer.

The RISC OS Desktop provides an ideal environment for using the Archimedes, but tends to get in the way when developing programs. For example, the only way to edit, or even examine, a Basic program is to temporarily leave the Desktop environment. The utility presented here helps to get around this by allowing the quick conversion of a Basic program into its text equivalent and vice versa. The resulting text file can then be edited using *Edit*.

ENTERING THE PROGRAM

Start by creating a new directory on disc called *B_to_T*, and go into this directory. Enter, save, and run listing 1, which will create and save a *!Sprites* file. Then use Edit to enter the Obey file from listing 2 and save this in the directory with the name *!Run*. Finally, enter listing 3 and save this in the directory as *BConvert*.

USING THE PROGRAM

The converter is installed from the Desktop by double clicking on its icon in the viewer. This will cause the icon to appear on the icon bar. The converter cannot be 'opened' as such - instead, files are dragged onto its icon. To convert a file, drag it onto the icon. After a short delay, a save box will pop up with the converted text file. This can either be saved, or dragged directly into another application. For example, to examine and modify a Basic program, drag its icon onto the converter. Then, drag the resulting text file into *Edit*. After making any changes, bring up *Edit*'s save box, drag the file back to the converter, and finally save the new Basic version of the file.

To merge two Basic programs, convert the first one and drag it into *Edit*. Position *Edit*'s text caret on the first blank line after the program listing. Next, convert the second program and drag its text file into the same *Edit* window. This will cause *Edit* to append the two text files. However, it is likely that the line numbers will not be correct. To get around this, just delete the line number from the first line. This will force the converter to renumber the program when it is converted back to Basic. Finally, 'save' the file from *Edit* into the converter, and then save the resulting Basic program which will contain the two original listings appended. Obviously, this can be extended to append any number of programs.

This month's magazine disc contains an enhanced version of the converter which allows a program to be edited outside the Desktop. To do this, hold down Shift while dragging the file onto the converter. Instead of opening a save box, Basic will be entered with the program present. After any changes have been made (possibly using the Basic Editor), press function key F1. This will re-enter the Desktop, and pop up a save box for the modified program. If you don't wish to save it, simply press Return to cancel the box.

Listing 1

```
   10 REM >SpriteMake
   20 DIM sp 1000:!sp=1000:sp!4=0
   30 sp!8=16:sp!12=16
   40 SYS "OS_SpriteOp",271,sp,"!B_to_T"
,0,34,17,12
   50 FOR Y%=0TO16:FOR X%=0TO33
   60 SYS "Wimp_SpriteOp",41,,"file_ffb"
,X%,Y% TO ,,,,,C%
   70 IF C%=7 THEN C%=8
   80 IF X%>Y%*2 THEN
   90 IF C%=0 THEN C%=7
  100 IF C%=8 THEN C%=0
  110 ENDIF
  120 SYS "OS_SpriteOp",298,sp,"!B_to_T"
,X%,Y%,C%
  130 NEXT:NEXT
  140 SYS "OS_SpriteOp",268,sp,"!Sprites
"
```

Listing 2

```
WimpSlot -min 32K -max 32K
IconSprites <Obey$Dir>.!Sprites
Run <Obey$Dir>.BConvert
```

Listing 3

```
   10 REM             >BConvert
   20 REM Program     Basic Converter
   30 REM Version     A1.00
   40 REM Author      David Spencer
   50 REM RISC User   July/August 1989
   60 REM Program     Subject to Copyright
   70 :
```

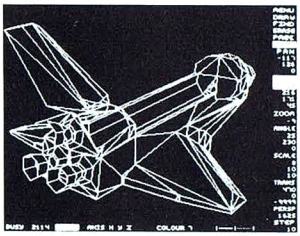
```
   80 DIM block 256,menu 50,oldblk 100
   90 P%=block:[OPT 0:STR R14,[R0]
  100 MOV PC,R14:]A%=block+20:CALL block
  110 match=block!20+72:taddr=match+4
  120 PROCun(match):PROCun(taddr)
  130 PROCassemble
  140 $block="TASK":SYS "Wimp_Initialise
",200,!block,"Basic<>Text" TO ,us
  150 file=FNwindow:DIM oks 2:$oks="OK"
  160 ok=FNicon(file,192,-160,238,-112,&
C701913D,oks,0,3)
  170 DIM fname 255,valid 3,sprite 20
  180 $valid="a~ "
  190 name=FNicon(file,10,-160,186,-112,
&700F13D,fname,valid,256)
  200 ficon=FNicon(file,78,-92,146,-20,&
D7006102,sprite,1,20)
  210 DIM sprite2 20:$sprite2="!B_to_T"
  220 sicon=FNicon(-1,0,0,68,68,&3102,sp
rite2,1,20)
  230 quit=FALSE:myref=-1
  240 ON ERROR PROCerror
  250 REPEAT
  260 SYS "Wimp_Poll",1,block TO reason
  270 CASE reason OF
  280 WHEN 2:SYS"Wimp_OpenWindow",,block
  290 WHEN 6:PROCbuttons(block)
  300 WHEN 7:PROCdragdone
  310 WHEN 8:PROCkey(block!24)
  320 WHEN 9:quit=TRUE
  330 WHEN 17,18:PROCreceive(block)
  340 WHEN 19:PROCnoack(block)
  350 ENDCASE
  360 UNTIL quit
  370 SYS "Wimp_CloseDown"
  380 END
  390 :
  400 DEF PROCreceive(b)
  410 CASE b!16 OF
  420 WHEN 0:quit=TRUE
  430 WHEN 1:PROCsavetous(b)
  440 WHEN 2:PROCdatasave(b)
  450 WHEN 3:PROCdataload(b)
  460 WHEN 6:PROCramfetch(b)
  470 WHEN 7:PROCramtransmit(b)
  480 ENDCASE
  490 ENDPROC
  500 :
  510 DEF PROCkey(k)
  520 CASE k OF
  530 WHEN 13:PROCsaveit($fname)
  540 WHEN 27:!block=file:SYS "Wimp_Clos
eWindow",,block:PROCrelease(cst)
  550 OTHERWISE:SYS "Wimp_ProcessKey",k
  560 ENDCASE
  570 ENDPROC
  580 :
  590 DEF PROCbuttons(b)
  600 IF (b!8 AND 2) THEN
  610 IF b!12=-2 PROCmenu(b)
  620 ELSE
  630 IF b!16=ficon AND (b!8 AND &50)<>0
 THEN
  640 !b=file
  650 SYS "Wimp_GetWindowState",,b
  660 wex=b!4-b!20:wey=b!16-b!24
  670 b!4=ficon
  680 SYS "Wimp_GetIconState",,b
  690 !b=file:b!4=5
  700 b!8=b!8+wex:b!12=b!12+wey
  710 b!16=b!16+wex:b!20=b!20+wey
  720 b!24=0:b!28=0
  730 b!32=&7FFFFFFF:b!36=&7FFFFFFF
  740 SYS "Wimp_DragBox",,b
  750 ENDIF
  760 IF b!16=ok THEN PROCsaveit($fname)
  770 ENDIF
  780 ENDPROC
  790 :
  800 DEF PROCmenu(b)
  810 $menu="Basic<>Text"
  820 menu!12=&70207:menu!16=156
  830 menu!20=40:menu!24=0:menu!28=&80
  840 menu!32=-1:menu!36=&7000021
  850 $(menu+40)="Quit"
  860 SYS "Wimp_CreateMenu",,menu,!b-64,
136
  870 ENDPROC
  880 :
  890 DEF FNwindow
  900 !block=900:block!4=200
  910 block!8=1146:block!12=368
  920 block!16=0:block!20=0
  930 block!24=-1:block!28=&93
  940 block!32=&1070207:block!36=&20D04
  950 block!40=0:block!44=-1024
  960 block!48=1280:block!52=0
  970 block!56=&3D:block!60=0
  980 block!64=1:block!68=0
  990 block!84=0:$(block+72)="Save as:"
 1000 SYS "Wimp_CreateWindow",,block TO
handle
 1010 =handle
 1020 :
 1030 DEF FNicon(window,x1,y1,x2,y2,flag
s,i1,i2,i3)
 1040 !block=window:block!4=x1
 1050 block!8=y1:block!12=x2
 1060 block!16=y2:block!20=flags
 1070 block!24=i1:block!28=i2
```

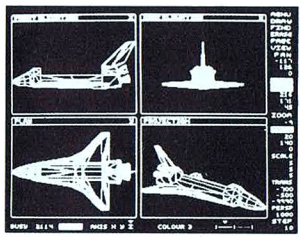
```
 1080 block!32=i3
 1090 SYS "Wimp_CreateIcon",,block TO ha
ndle
 1100 =handle
 1110 :
 1120 DEF PROCdragdone
 1130 IF totext THEN type=&FFF ELSE type
=&FFB
 1140 SYS "Wimp_GetPointerInfo",,block
 1150 block!20=64:block!32=0
 1160 block!36=1:block!40=block!12
 1170 block!44=block!16:block!48=!block
 1180 block!52=block!4
 1190 block!56=end-bst:block!60=type
 1200 $(block+64)=FNleaf($fname)+CHR$0
 1210 SYS "Wimp_SendMessage",17,block+20
,block!12,block!16
 1220 myref=block!28:start=bst
 1230 ENDPROC
 1240 :
 1250 DEF FNleaf(path$)
 1260 WHILE INSTR(path$,"."):path$=MID$(
path$,INSTR(path$,".")+1)
 1270 ENDWHILE
 1280 =path$
 1290 :
 1300 DEF PROCerror
 1310 SYS "Wimp_DragBox",,-1
 1320 !block=ERR:$(block+4)=REPORT$+" (I
nternal error code "+STR$ERL+")"+CHR$0
 1330 SYS "Wimp_ReportError",block,1,"Ba
sic<>Text"
 1340 ENDPROC
 1350 :
 1360 DEF PROCsavetous(b)
 1370 totext=(b!40=&FFB)
 1380 FOR F%=0 TO 43 STEP 4
 1390 oldblk!F%=b!F%:NEXT
 1400 b!12=b!8:b!16=6
 1410 ff$=FNgname(b+44)
 1420 bst=FNclaim(b!36*2)
 1430 bend=bst+b!36*2:bst2=bst+b!36
 1440 b!20=bst2:b!24=bend-bst2
 1450 SYS "Wimp_SendMessage",17,b,b!4
 1460 myref=b!8:retry=TRUE:size=0
 1470 ENDPROC
 1480 :
 1490 DEF PROCnoack(b)
 1500 IF b!12=myref THEN
 1510 IF retry THEN
 1520 oldblk!12=oldblk!8:oldblk!16=2
 1530 !oldblk=100
 1540 $(oldblk+44)="<Wimp$Scrap>"+CHR$0
 1550 SYS "Wimp_SendMessage",17,oldblk,o
ldblk!4
 1560 myref=oldblk!8:retry=FALSE
 1570 ELSE ERROR 1,"Pipe Broken"
 1580 ENDIF
 1590 ENDIF
 1600 ENDPROC
 1610 :
 1620 DEF PROCdatasave(b)
 1630 IF b!12=myref THEN
 1640 PROCsaveit(FNgname(b+44))
 1650 b!12=b!8:b!16=3
 1660 SYS "Wimp_SendMessage",18,b,b!4
 1670 ENDIF
 1680 ENDPROC
 1690 :
 1700 DEF PROCramtransmit(b)
 1710 size+=b!24
 1720 IF b!24=bend-bst2 THEN
 1730 b!12=b!8:b!16=6
 1740 SYS "Wimp_SendMessage",18,b,b!4
 1750 ELSE end=FNconvert(bst2,size,bst)
 1760 PROCpopup
 1770 ENDIF
 1780 ENDPROC
 1790 :
 1800 DEF PROCdataload(b)
 1810 IF b!40=&FFF OR b!40=&FFB THEN
 1820 totext=(b!40=&FFB)
 1830 ff$=FNgname(b+44):X%=OPENIN ff$
 1840 s%=EXT#X%:CLOSE #X%
 1850 bst=FNclaim(s%*2):bend=bst+s%*2
 1860 SYS "OS_File",&FF,ff$,bst+s%
 1870 end=FNconvert(bst+s%,s%,bst)
 1880 PROCpopup
 1890 IF b!12=myref THEN *DELETE <Wimp$S
crap>
 1900 b!12=b!8:b!16=4
 1910 SYS "Wimp_SendMessage",17,b,b!4
 1920 ENDIF
 1930 ENDPROC
 1940 :
 1950 DEF PROCramfetch(b)
 1960 IF b!24<cnd-start THEN cend=start+
b!24 ELSE cend=cnd
 1970 SYS "Wimp_TransferBlock",us,start,
b!4,b!20,cend-start
 1980 b!12=b!8:b!16=7:b!24=cend-start
 1990 SYS "Wimp_SendMessage",18,b,b!4
 2000 start=cend
 2010 IF start=cnd THEN !menu=file:SYS "
Wimp_CloseWindow",,menu:PROCrelease(cst)
 2020 ENDPROC
 2030 :
 2040 DEF FNgname(ptr):LOCAL f$
 2050 WHILE ?ptr f$+=CHR$?ptr:ptr+=1
 2060 ENDWHILE
```

SILICON VISION

SOFTWARE FOR THE ARCHIMEDES & BBC

SolidCAD





The ultimate 3D Draughting System for Architectural design, Interior design, Engineering Design and Teaching CDT. Allows drawing in plan, front & side elevations and also directly in 3D view. Includes powerful zoom & pan options for precision draughting and surface definition for creating solid colour objects. Also includes Sweep, Extrude & Macro facilities for designing very complex objects easily. Designs created with SolidCAD are compatible with the Realtime Graphics Language for high-speed flicker-free animation. The custom Archimedes version also performs smooth shading for realism. SolidCAD(Arc) users can upgrade to the Realtime Solids Modeller (Arc) for £40.00.

£49.95 (ARC or BBC B/B+/Master),

REALTIME SOLIDS MODELLER

The package includes both the sophisticated design environment of SolidCAD and the high speed animation capability of a Realtime Graphics Language (RGL) module developed in pure ARM Risc code for supercharged performance. The package is ideal for Architectural design, Interior design, Engineering design & teaching CDT. The RGL module can be used to create standalone flicker-free animation of designs from your own programs. Smooth shading is also performed for realistic images. Through our in-house expertise in 3D Design and High-speed techniques, no other package can rival the design environment & animation speed of the Realtime Solids Modeller.

£89.95 (ARC)

REALTIME GRAPHICS LANGUAGE

The Realtime Graphics Language rom provides a complete 3D Solids/Wireframe animation system with 52 star commands and 3D Editors for designing objects to animate from your own programs. Includes a 35,000 pixels / sec line generator for fast 3D drawing rates. 3D Rotate, Scale, Orbit, Perspective and Turtlegraphics. Also compatible with designs created with SolidCAD (BBC).

£49.95 (BBC B/B+/Master)

SUPER-DUMP

The ultimate printer driver which takes advantage of the highest resolution capability of ordinary Epson compatible printers to provide 1920 × 1024 resolution. Images can also be scaled, positioned and previewed before printing. Fully compatible with SolidCAD, Realtime Graphics Language, Gate-Array design system & 3D CAD/Animation system. Your own graphics programs or other CAD packages can be made compatible with Super-Dump by the addition of a few simple commands. An example program is included in the package.

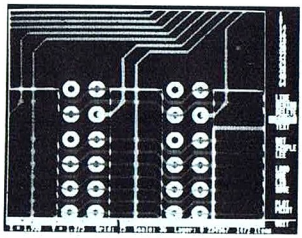
£15.95 (BBC B/B+/Master), £24.95 (ARC)

Presentation System

The package provides an interactive environment to create, edit and play-back computer controlled presentations for lectures and demonstrations. Also handles graph plotting for polynomials & user-defined functions which can be incorporated within the presentations.

£34.95 (BBC B/B+/Master), £49.95 (ARC)

ARC-PCB



The ultimate PCB design system developed specifically for the Archimedes with a specification that cannot be matched. Includes Automatic routing, Rats-nesting, 8 layers, Surface mount capability, 0.001" resolution, 32" × 32" maximum board size, On-line Help, Fast Zoom/Pan/Redraw, Text & Silkscreen facility, Variable Line/Pad/Text/Grid sizes, Part Libraries, Block Move/Copy/Rotate/Mirror/Erase options, and up to 300,000 components.

For hardcopy, the system supports the industry standard HP-GL, GRAPHTEC, PLOTMATE plotters and printers at their highest resolutions. Also provides automatic drilling information with extensive support from PCB manufacturers for final layout and production, optional 1 year telephone Hotline support (£100) and software maintenance service (£75) is also available.

£195.00 (ARC)

Risc BASIC

The first true BASIC V syntax compiler which will convert your programs into supercharged Risc code for turbo performance leaving the competition standing. Features include Relocatable modules, full cross references, Double precision floating point & Integer support, in-line assembly, Window-based or command line compilation environment, standalone code generator, object code optimiser, full array manipulation and dimensions support, multiple exit structures & full runtime error handler.

£99.95 (ARC)

RiscFORTH

A new 32-bit implementation of the FORTH-83 standard, designed to take full advantage of the ARM architecture. Features include Multi-tasking, Optimising compiler, built-in ARM assembler with floating point mnemonics, built-in Full screen Editor, File system interface, OS calls support, Floating point & Integer maths, WIMP support, Single-step debugger, Shadow screen for documentation, Block manipulation, Dictionary & Vocabulary display, Call finding and a standalone code generator.

£99.95 (ARC)

All Archimedes software run in native mode on A305 – A440 & A3000 with Arthur 1.2 or RISC OS

SILICON VISION LTD, SIGNAL HOUSE, LYON ROAD, HARROW MIDDLESEX HA1 2AG. TEL: 01-422 2274 or 01-861 2173
FAX: 01-427 5169. TELEX: 918266 SIGNAL G.

(Access/Mastercard/Eurocard accepted)
All prices include VAT and Carriage (Overseas orders add £4).

```
 2070 =f$
 2080 :
 2090 DEF PROCsaveit(name$)
 2100 IF INSTR(name$,".") THEN
 2110 IF totext THEN type=&FFF ELSE type
=&FFB
 2120 SYS "OS_File",10,name$,type,,cst,c
nd
 2130 PROCrelease(cst):!menu=file
 2140 SYS "Wimp_CloseWindow",,menu
 2150 ELSE $block="    To save, drag the
 icon to a directory viewer"
 2160 SYS "Wimp_ReportError",block,1,"Ba
sic<>Text"
 2170 ENDIF
 2180 ENDPROC
 2190 :
 2200 DEF PROCpopup
 2210 cst=bst:cnd=end:fn$=FNleaf(ff$)
 2220 fm$=LEFT$(ff$,LENff$-LENfn$)
 2230 IF LENfn$=10 THEN fn$=LEFT$(fn$,9)
 2240 IF totext fp$="T" ELSE fp$="B"
 2250 $fname=fm$+fn$+fp$:!oldblk=file
 2260 IF totext THEN $sprite="file_fff"
ELSE $sprite="file_ffb"
 2270 SYS "Wimp_GetWindowState",,oldblk
 2280 SYS "Wimp_OpenWindow",,oldblk
 2290 SYS "Wimp_SetCaretPosition",file,n
ame,,,-1,LEN$fname
 2300 ENDPROC
 2310 :
 2320 DEF FNclaim(size):SYS "OS_Module",
6,,,size TO ,,ptr:=ptr
 2330 :
 2340 DEF PROCrelease(RETURN ptr):IF ptr
 THEN SYS "OS_Module",7,,ptr:ptr=0
 2350 ENDPROC
 2360 :
 2370 DEF PROCun(RETURN addr)
 2380 addr=addr+8+(!addr AND &FFFFFF)*4
 2390 ENDPROC
 2400 :
 2410 DEF FNconvert(from,len,to)
 2420 SYS "Hourglass_On"
 2430 IF totext THEN
 2440 D%=from+1:E%=to:to=USR code
 2450 ELSE
 2460 A%=from:B%=from+len:C%=to+1
 2470 ?to=13:to=USR tok
 2480 ENDIF
 2490 SYS "Hourglass_Off"
 2500 =to
 2510 :
 2520 DEF PROCassemble
 2530 DIM code 512
 2540 FOR pass=0 TO 2 STEP 2
 2550 P%=code:[OPT pass
 2560 STMFD R13!,{R12,R14}
 2570 .l LDRB R1,[R3,#1]
 2580 LDRB R0,[R3]:ADD R0,R1,R0,LSL#8
 2590 CMP R0,#&FF00:MOVCS R0,R4
 2600 LDMCSFD R13!,{R12,PC}^
 2610 MOV R1,R4:MOV R2,#6:ADD R3,R3,#3
 2620 SWI "OS_ConvertCardinal2"
 2630 MOV R4,R1:.l1 LDRB R0,[R3],#1
 2640 CMP R0,#13:BEQ ld:CMP R0,#&8D
 2650 BEQ ln:CMP R0,#&7F
 2660 STRCCB R0,[R4],#1:BCC l1
 2670 MOV R12,R3:BL taddr:MOV R3,R12
 2680 .l2 LDRB R0,[R1],#1:CMP R0,#&7F
 2690 STRCCB R0,[R4],#1:BCC l2:B l1
 2700 .ld MOV R0,#10:STRB R0,[R4],#1
 2710 B l:.ln LDRB R0,[R3],#1
 2720 MOV R2,R0,LSL #2:LDRB R0,[R3],#1
 2730 AND R1,R2,#&C0:EOR R0,R0,R1
 2740 MOV R5,R0:LDRB R0,[R3],#1
 2750 MOV R1,R2,LSL #2:EOR R0,R0,R1
 2760 AND R0,R0,#&FF:MOV R2,#6
 2770 ORR R0,R5,R0,LSL #8:MOV R1,R4
 2780 SWI "OS_ConvertCardinal2"
 2790 MOV R4,R1:B l1
 2800 .tok STMFD R13!,{R0-R2,R14}
 2810 MOV R6,R0:MOV R7,R1:MOV R8,R2
 2820 MOV R9,#0:.t1 CMP R6,R7
 2830 ADDCS R13,R13,#4:MOVCS R0,#&FF
 2840 STRCSB R0,[R8],#1:MOVCS R0,R8
 2850 LDMCSFD R13!,{R1-R2,PC}
 2860 LDRB R0,[R6],#1
 2870 CMP R0,#13:CMPNE R0,#10
 2880 CMPNE R0,#32:BEQ t1:SUB R1,R6,#1
 2890 CMP R0,#&30:BCC nono:CMP R0,#&3A
 2900 BCC num:.nono CMP R9,#0
 2910 MOVEQ R9,#10:LDMEQFD R13,{R6-R8}
 2920 BEQ t1:MOV R2,R9:ADD R9,R9,#10
 2930 B num2:.num LDR R2,llim
 2940 LDR R0,base:SWI "OS_ReadUnsigned"
 2950 CMP R9,#0:BNE nono
 2960 .num2
 2970 STRB R2,[R8,#1]:MOV R2,R2,LSR #8
 2980 STRB R2,[R8]:ADD R2,R8,#3
 2990 MOV R3,#0:MOV R4,#0:BL match
 3000 MOV R6,R1:SUB R0,R2,R8
 3010 STRB R0,[R8,#2]:ADD R8,R8,#3
 3020 .elc LDRB R0,[R8],#1:CMP R0,#32
 3030 BEQ elc:CMP R0,#&8B:MOVEQ R0,#&CC
 3040 STREQB R0,[R8,#-1]:MOV R8,R2:B t1
 3050 .base EQUD 10+(1<<29)
 3060 .llim EQUD &FEFF:]NEXT
 3070 ENDPROC
```

ANCESTRY from MINERVA

Reviewed by Sheridan Williams

INTRODUCTION

Tracing your family history can be a fascinating experience, and one that I'm sure many more people would get involved with if it were not for the sheer amount of paperwork needed. Ancestry, to a large extent, takes the drudgery out of record keeping. So if you are interested in starting your own family tree research, but don't know where to start, buy Ancestry, enter all the information that can be gleaned from your living relatives, and lo and behold, you'll have the basis from which to expand. Starting this way, with a package as sophisticated as Ancestry you will quickly find that you are hooked on finding out more. Genealogy is like doing a jigsaw - each extra piece of information helps complete the picture, except that a family tree can never be complete however far back you go, but that simply adds to the intrigue.

THE PACKAGE

Ancestry comes in a distinctive mauve and yellow standard plastic case, with 56 page manual, disc and registration slip, the latter explaining that you get 90 days free software support, and inviting you to a further year's support for £30. The manual is adequate, but I found it vague in places, with the index not directing you to all the references. It explains that you should first make two backup copies of the original disc, unfortunately Minerva's method of preventing unauthorised use, means that you will always need to keep the original disc to hand, as you will be prompted to insert it whenever you leave the title screen. This applies also to copies made to hard disc. Minerva suggest that you contact them if you are unable to live with this method of operation.

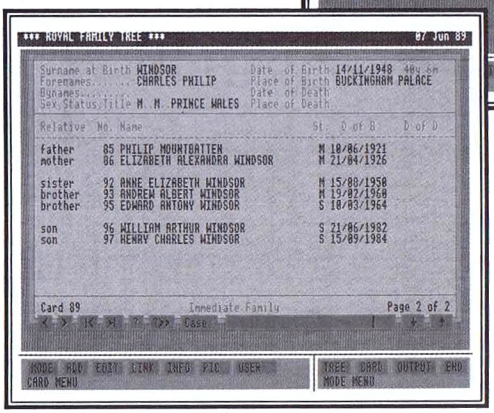


Figure 1. Tree display of Prince Charles' immediate family

FEATURES

Minerva supply the British Royal Family from Henry VII (born 1457) to Prince Henry (born 1984) as a sample data file. Section 3 of the manual takes you through the Royal Family file in a tutorial fashion, and provides a superb way of getting to know the package in

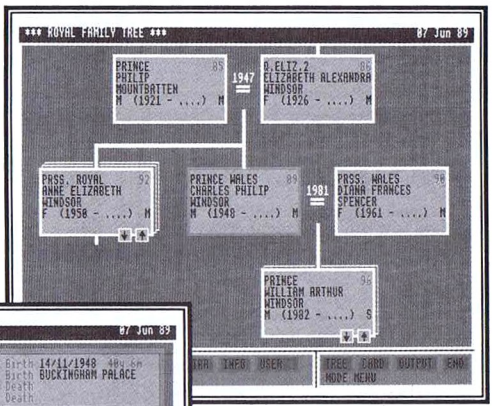


Figure 2. Prince Charles' on-screen record card

detail. You get a real feel for just how powerful Ancestry is. Also supplied are Royal Family *User* and *Picture* files containing notes about, and pictures of individuals. However, if you wish to include pictures in your own file this will require a digitiser, or art package for their generation. The pictures can be displayed along with other details. Having familiarised yourself with the Royal Family file, creating your own data file is a relatively straightforward process.

Two modes of browsing are available, TREE or CARD. TREE displays one generation before and after the individual currently selected (at startup this is individual number 1), and clicking on any person will display the three immediate generations to

which they are related. Clicking on the current individual switches to card mode for that person, and shows in three pages marriages and immediate family. Both modes allow fast movement around the file, including a search for matches.

CREATING YOUR OWN FILE

This is simply achieved by simply adding cards to an otherwise empty database, you need not concern yourself with anything else. Matters become slightly more complex if you require the optional "User" file which can contain free format text about individuals, in which case you have to "Create Skeleton User File" and decide at the outset how many cards you want.

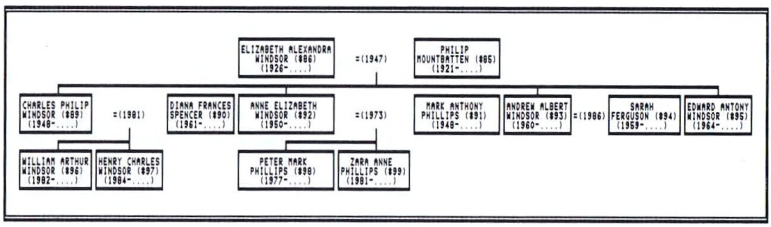


Figure 3. Family tree for Queen Elizabeth II

As you enter the details about an individual such as forenames, surname, dates and places of birth and death, sex, status and title, the program allocates it a unique reference number. You could use this reference for labelling photographs, documents etc. The next task is to "link" various people in either of two ways - partner or child. The former need not be a marriage, and a person may have up to 7 partners. Having formed a partner link, you can now link offspring. Warnings can appear at this stage for example "<16 at date of marriage", "Child born after first parent's death", which I found very useful in trapping errors.

OUTPUTS

The package supports various standard reports - for Individuals, Ancestors and Descendants, any printer will do for producing these. There is also a very powerful and easy-to-use Report Generator, allowing you to print reports of your own design.

TREE PRINTING

For many, producing the family tree is the piece de resistance, and something that several packages fail to do automatically. Amazingly, if your tree will not fit a single sheet, the package will number the sheets for you to paste together. Printout takes place sideways and requires an Epson compatible printer. You can decide on any of tiny, small, medium or large print sizes, the smallest allowing 12 generations per page. You can also select light or dark print (the latter taking twice as long as it uses double strike mode).

CONCLUSION

Unfortunately for those (like me) who would like to transfer data from other software, Ancestry is let down by its inability to generate or accept data in the internationally accepted GEDCOM data format. However, there is a facility to export data in CSV (comma separated value) format. I approached Minerva about getting data into the package, and their response was that the file design was extremely complex, and other than approaching them to write a transfer routine (for which they would charge) the only possibility is to examine the "add a card" routine in the Basic program and try and patch in the transfer yourself. I succeeded in modifying PROCadd (lines 2760-3200) in program Anc/161, and in particular I replaced line 2940 with the necessary input routine.

Apart from the data transfer aspect, I find it difficult to fault Ancestry, and conclude that it is a brilliant package. If you can find any excuse to buy it, then I suggest that you do so immediately. But beware - even though I have been tracing my family for many years I found I could not put Ancestry down.

Product	**Ancestry**
Supplier	**Minerva Systems** **69 Sidwell Street, Exeter,** **Devon EX4 6PH.** **Tel. (0392) 437756**
Price	**£79.95 inc. VAT**

RU

TimeWatch

TimeWatch provides the tools to manage appointments, a 'To Do' list of tasks, and jogs the memory for those important birthdays and anniversaries. TimeWatch retains your information in pages – there is a page for each day of the year, for as many years as desired.

Appointments. Appointments for the day are detailed by start time, estimated duration and a descriptive note. All appointments for the day are displayed in time sequence.

Short Notes. Useful for jotting down a quick reminder, or notes of a telephone conversation.

Tasks to do. TimeWatch will help you plan your time and keep track of outstanding work by displaying your details of tasks in one of the categories 'Start Today', 'Ongoing' or 'Overdue!'.

Birthdays, anniversaries etc. Don't forget those things that should not be forgotten. Details entered in this category will appear on the same page for all years.

Mitre Software Limited
International House
26 Creechurch Lane
London EC3A 5BA
Tel: 01-283 4646

TimeWatch can also search for a piece of text in any, or all, of your categories of information; print some, or all, of the information stored in a range of pages; and, by means of the perpetual calendar, move quickly from one page to another.

For those who have made use of the Arthur 1.2 DeskTop Diary, TimeWatch will import the contents of the diary files.

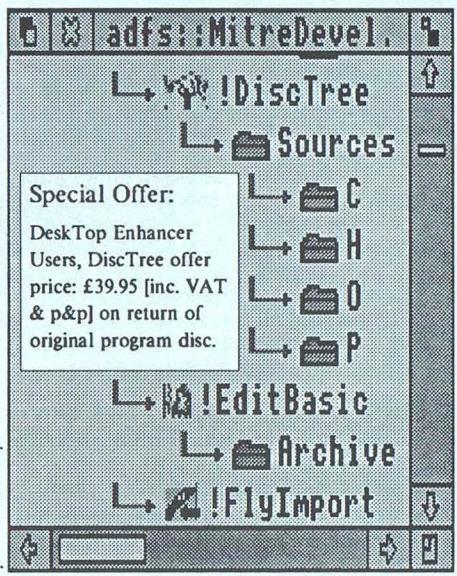
DiscTree

provides comprehensive, and easy to use, facilities to display the directory structure of a disc, search for files and backup files.

A TreeViewer provides the means to move rapidly around the directory structure of a disc and search for files. After specifying the directories to search, name and type to match the files found will be displayed, ready for loading into an editor.

A backup is performed after specifying the directories to search, file name, type and date to match. These criteria may be saved to a script file for fast and easy repeated use.

Backup copies files matching your criteria to floppy discs. Files are spread over many discs if required. Restore displays the contents of backup discs and allows the restoration of files by dragging to the required destination, or automatic restoration to their original locations.



DiscTree and TimeWatch are written specifically for RISC OS desktop and make full use of the WIMP, multi-tasking and inter application communication.

Mitre Software Limited

Prices: DiscTree: £49.95. TimeWatch: £29.95 [inc. VAT and p&p].

Into the ARC

Mike Williams discusses the use of functions and procedures in Basic V.

Last month, in the first of this new series, I took a broad view of the whole process of program development. If your acquaintance with programming, particularly using BBC Basic on the Archimedes is rather limited, then you maybe felt that I left too many questions unanswered. Well, I only have so much space in the magazine, so this month I propose to look in more detail at some of those features which I had to refer to briefly last time.

Foremost among the better things of BBC Basic are *functions* and *procedures.* If you are going to follow the structured approach to your programming which I advocated last time, then you need to use procedures and functions from the word go. To some this may seem like trying to run before you can walk, but I firmly believe that if you get the bones of your program right, filling in the detail becomes a lot easier.

Functions and procedures both work on the same principle. We identify some small routine or task within a program, we package it up with a unique name, and then whenever we need to use that routine we just refer to it by the name previously bestowed. Putting it more technically, we need to *define* a function or procedure before it can be *called.* However, provided the definition of the function or procedure exists somewhere in the program, Basic is not that fussy as to where it is located. So whether the definition or the call appears nearer the start of the program doesn't matter.

Traditionally, in BBC Basic, procedure and function definitions tend to form the latter part of the program, and calls to these routines usually appear earlier. When a procedure or function is first called, Basic searches through the program till it finds the appropriate definition. It then enters the name of the procedure or function in a table so that the next time the same routine is called it can be found much more quickly by looking up its name in the table.

And what, you may be wondering, is the difference between a function and a procedure. It's quite simple: a function always returns an answer or result, whereas a procedure performs a task which may or may not produce any 'results'. A user-defined function is like any other in Basic, such as SIN or ABS for example, and may be used similarly. A procedure is more like an instruction (for example, PRINT), and is used in similar situations, though it doesn't have to have anything to do with printing of course.

To define a procedure we must write:

```
DEF PROCname
.................
<body of procedure>
.................
ENDPROC
```

but you can give your procedures any names which you choose (using the characters 0-9, A-Z, a-z, '£' and 'underline'), and it usually helps to give meaningful names that remind you of the procedure's purpose. Typically in a program you might see such procedures as:

```
PROCset_up
PROCtitle
PROCmenu
PROCselect
PROCreport1
etc.
```

The so-called *body* of the procedure simply consists of whatever instructions are needed to perform the task in question. We might have a procedure definition such as:

```
1000 DEF PROCset_variables
1010 x1=0:x2=640:y1=0:y2=512
1020 exit=FALSE:repeat=TRUE
1030 ENDPROC
```

Look at the procedures used in programs published in RISC User. These will give you plenty of examples to study and follow, and you may find some that can be adapted to your own programs. A procedure is called by simply writing its name (preceded by the four letters PROC).For example, the procedure defined above would probably be called once near the start of the program such as:

```
150 PROCset_variables
```

Now the procedure *PROCset_variables* is completely self-contained, requiring no other information from the outside world (i.e. from

the rest of the program). That's fine, but often when a procedure is called, we need to pass it some additional information. For example, it can be very useful to have a procedure which will display a text message at any point on the screen, and in a colour of our choice. For the procedure to work it will need to know the x and y co-ordinates of the screen position, the text to be displayed, and the colour to use. Such pieces of information passed to a procedure are called parameters, and are included in the procedure definition.

Thus we might write:

```
1100 DEF PROC display_text(x,y,msg$,c)
1110 COLOUR c
1120 PRINTTAB(x,y)msg$;
1130 ENDPROC
```

When the procedure is called, we specify the parameters we want. So, for example:

PROCdisplay_text(5,1,"Hello world",6)

would display the message "Hello world" in cyan (colour 6 in any 16 colour mode) starting in position 5 on line 1. You can try this out directly, once you have type in the procedure definition, to test that it works.

The parameters '5', '1', etc. are called the *actual* parameters, because they are the ones really used, whereas the parameters 'x', 'y', etc. used in the definition are called the *formal* parameters because they exist only to allow us to define formally the procedure in the first place. Formal parameters can be thought of as variables that only exist inside the procedure definition. Furthermore, Basic never confuses the use of 'x' say as a formal parameter, and the use of 'x' as a variable anywhere else in the program. In effect, a formal parameter is *local* to a procedure.

Now let's consider another example, this time a procedure to get the name of a month of the year given the month number, and print it out. Here is the procedure definition:

```
1200 DEF PROCget_month(month_no)
1210 FOR I%=1 TO month_no
1220 READ month$
1230 NEXT I%
1240 PRINT month$
1250 ENDPROC
1260 DATA January, February, March
1270 DATA April, May, June, July
1280 DATA August, September, October
1290 DATA November, December
```

To print out the name of the sixth month of the year, for example, we could write:

```
110 PROCget_month(6)
```

The procedure has one parameter, the number of the required month. In addition, the procedure needs to use an extra variable *(I%)* to control the FOR-NEXT loop, and a variable *month$* to temporarily store the name of a month. When writing many procedures in a program, it can be difficult to remember whether any such particular variable is already in use for another purpose. We can avoid any potential problems by *declaring* any extra variables used in this way as being *LOCAL* to the procedure. This means that Basic will correctly distinguish between any variable declared in this way, and the same variable name used elsewhere in the program. We will do this for *PROCget_month* above by adding:

```
1205 LOCAL I%,month$
```

If that line were included, Basic would remember any value assigned to *I%* on entry to the procedure, and restore it on exit.

You can easily test this. Take the modified procedure above and add:

```
100 I%=-999
110 PROCget_month(6)
120 PRINT I%
130 END
```

to form a complete program. Run this and you will see the value of *I%* printed as -999 even though the procedure has used *I%* for quite a different purpose.

It is always sensible to declare as LOCAL all variables (and arrays, which we'll deal with on another occasion) which are introduced and used only within the definition of a procedure. If you want to send any information to a procedure, do so as a parameter. That is the 'good' approach, and with longer, more complex programs it does pay dividends.

Many programmers are quite casual about the use of LOCAL, frequently omitting it unless absolutely essential, and often their programs still work. Trouble is more likely to arise when

modifications are made to a program that has been in existence for some time. If you have not used LOCAL, it is only too easy to inadvertently pick on a variable to use in a procedure definition that has some quite different purpose elsewhere in the program. If clashes do occur, the results can be very unpredictable, and difficult to trace.

So the general rule is to use parameters to pass information to a procedure and to use LOCAL for everything within the procedure. However, with care we can be a little more relaxed than this, with some advantages as we shall see in the examples which follow.

In the case of the earlier procedure to print the name of a month given the number of the month, it seemed natural to use the month number as a parameter. In other examples it can seem just as natural to refer directly to some variable.

Here is a short procedure to read in a number of data items (the number is the parameter of the procedure), and to store these in a list or array. This is referred to directly, and is assumed to have been dimensioned earlier in the program.

```
500 DEF PROCread_data(n)
510 LOCAL I%
520 FOR I%=1 TO n
530 INPUT"Next value: " data(I%)
540 NEXT I%
550 ENDPROC
```

The procedure executes a loop to read in one value at a time, and store this in the array *data()*. The loop variable *I%* is only relevant inside the procedure so it has been declared as LOCAL.

All that I have said so far can be equally applied to functions. In addition, a function always returns at least one value. The definition of a function takes the form:

```
DEF FNname
.................
<body of function>
.................
=<value>
```

The name is preceded by the letters 'FN', and the definition terminates with an equals sign followed by an assigned value, but otherwise it follows very much the procedure model.

Suppose we now want to analyse the data we have read in to find the maximum, minimum and average of the stored numbers. A function is the ideal choice now:

```
600 DEF FNmax(n)
610 LOCAL I%,max:max=data(1)
620 FOR I%=2 TO n
630 IF data(I%)>max THEN max=data(I%)
640 NEXT I%
650 =max
```

This one finds the maximum value in the array. It sets the (temporary) maximum to be the first value in the array, and then compares this with all subsequent values, replacing it if a larger value is found. The last line of the definition returns that maximum value.

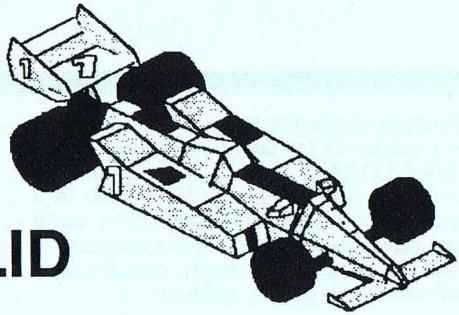
To see how the function is used, we could add some additional instructions to the definitions of *PROCread_data* and *FNmax* above to form a complete program:

```
100 MODE 12
110 INPUT"How many values": " n
120 PROCread-data(n)
130 PRINT"The maximum value is ";FNma
x(n)
```

Since the function returns a value, it can be used anywhere a value is used. Here it is included directly in a PRINT statement. You may like to try writing further functions to similarly find the minimum and average of the values read in.

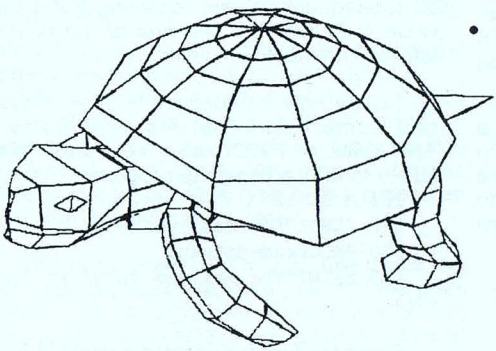
Functions, and particularly procedures, should form the background of any well-written program. Don't be afraid of using them, and as often as you like. Although you may feel that their main value is in avoiding the unnecessary repetition of the same sequence of instructions, you will find many examples where a procedure is called just once. Using procedures allows larger programs to be divided up into smaller and more manageable 'bites'. With this approach you can concentrate on getting each individual routine correct before moving on to the next. RU

Ace Computing

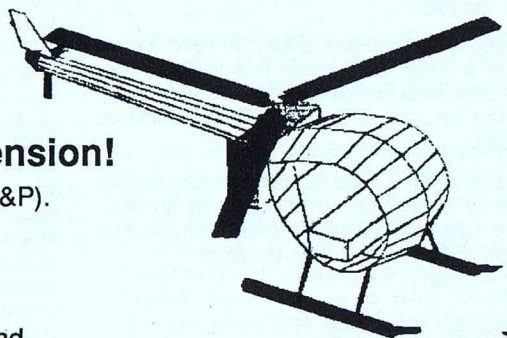


RISC OS EUCLID IS HERE!

You've got RISC OS – fun isn't it?
Here's something to stretch it...



- Multidocument 3-D Editor
- Simple, intuitive, icon driven, single window interface
- Object orientated data structure
- Curved surface primitives
- Cameras and lights
- Grid locking and measurement facilities
- Import and Export of Draw files



EUCLID
- Explore a new dimension!

Price: **£70** (inc VAT and P&P).

Available by mail order from:
Ace Computing, 27 Victoria Road,
Cambridge CB4 3BW.
Tel: (0223) 322559.
Or from your local dealer.

Ace Computing

A MULTI-TASKING DISC FORMATTER

Glynn Clements presents a program which allows you to format floppies in the background.

While the Desktop does allow floppy discs to be formatted, it does so in a rather crude way. In particular, all other applications are suspended for the duration of the operation. The program here allows a disc to be formatted in the background, meaning that you can continue with other applications while the formatting takes place. In addition, discs can also be verified in a similar way.

ENTERING THE PROGRAM

Start by creating a new directory called *!Format*, and select this as the current directory. Enter the Desktop, and run the *!Edit* application. Use the 'Create new Obey file' option (from the iconbar menu) to enter listing 1, and save this as *!Run*. Next, go to Basic and enter listing 2. Save this with the name *!Runimage*. This should leave the directory !Format containing the two files *!Run* and *!Runimage*.

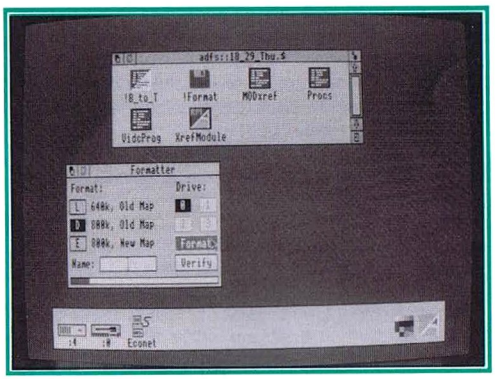


Figure 1. The formatter installed on the Desktop

You might wish to design a sprite which will be used to represent the formatter. To do this, use *Paint* to create a mode 12 sprite which is 34 by 17 in size with no palette and no mask. Call this sprite *!Format*, and save it in a file called *!Sprites* within the *!Format* directory.

USING THE FORMATTER

The formatter is started by double clicking on the *!Format* directory. This pops up the window shown in figure 1. The four 'buttons' at the top right allow the drive to be selected. It is only possible to select drives which are actually fitted to the machine, the other buttons being greyed-out. The three buttons down the left hand side select the format to use - L, D or E. The two large buttons at the bottom right start the format, or verify, options. In the case of formatting, a disc name must be entered into the box titled *Name* before the operation can begin. You are also asked for confirmation before formatting.

While actually formatting or verifying, the bar at the bottom of the window shows the progress, with red indicating formatting, and green verifying. A disc is automatically verified after it has been formatted. The menu button pops up a menu which offers the option to abort the current operation, or to quit the formatter. Clicking on the close box of the window will also quit the formatter. Be warned, though, the operation of the entire system is very slugish while the formatting is in progress.

Listing 1

```
Set Format$Dir <Obey$Dir>
WimpSlot -min 32K -max 32K
Run <Format$Dir>.!RunImage
```

Listing 2

```
   10 REM             >!RunImage
   20 REM Program     Background Formatter
   30 REM Version     A 1.0
   40 REM Author      Glynn Clements
   50 REM RISC User   July/August 1989
   60 REM Program     Subject to Copyright
   70 :
   80 DIM bk% 100,buffer% 56,spArea% 700
,data% 6500,rec% 64,dlist% 1024,icon_dri
ve%(3)
   90 SYS "Wimp_Initialise",200,&4B53415
4,"Formatter"
  100 ON ERROR PROCerror(ERL,REPORT$)
  110 spArea%!0=2000:spArea%!4=0
  120 spArea%!8=16:spArea%!12=16
  130 SYS"OS_SpriteOp",&10F,spArea%,"Bar
Graph",0,240,5,12
  140 FOR i%=0 TO 239
  150 FOR j%=0 TO 4
  160 SYS"OS_SpriteOp",&12A,spArea%,"Bar
Graph",i%,j%,0
  170 NEXT j%,i%
  180 PROCwindow
  190 SYS "Wimp_SetCaretPosition",handle
%,icon_name%,,,-1,0
  200 quit%=FALSE:quit_ok%=FALSE
```

```
  210 mode%=-1
  220 REPEAT
  230 SYS "Wimp_Poll",FNmask,bk% TO reas
on%
  240 CASE reason% OF
  250 WHEN 0:IF mode%>-1 THEN PROCdo_tra
ck(drive%,format$,mode%)
  260 WHEN 2:SYS "Wimp_OpenWindow",,bk%
  270 WHEN 3:PROCclose_down(FNquit_ok)
  280 WHEN 6:
  290 change%=bk%!8 AND (NOT bk%!20)
  300 IF(change% AND 5)ANDmode%=-1 THEN
  310 CASE bk%!16 OF
  320 WHEN icon_format%: PROCgo(0)
  330 WHEN icon_verify%: PROCgo(1)
  340 ENDCASE
  350 ENDIF
  360 IF (change% AND 2) THEN PROCmenu(b
k%!0-64,bk%!4)
  370 WHEN 8:
  380 IF !bk%=handle% AND bk%!4=icon_nam
e% AND mode%=-1 THEN
  390 IF bk%!24=13 THEN PROCgo(0) ELSE S
YS "Wimp_ProcessKey",bk%!24
  400 ENDIF
  410 WHEN 9:
  420 IF bk%!0=0 THEN
  430 PROCclose_down(FNquit_ok)
  440 ELSE
  450 mode%=-1:PROCupdate_bar(0,79,0)
  460 ENDIF
  470 WHEN 17,18:
  480 CASE bk%!16 OF
  490 WHEN 8:quit_ok%=FNquit_ok
  500 WHEN 0:PROCclose_down(quit_ok%)
  510 ENDCASE
  520 ENDCASE
  530 UNTIL quit%
  540 SYS "Wimp_CloseDown"
  550 END
  560 :
  570 DEF FNmask
  580 IF mode%=-1 THEN =&33 ELSE =&32
  590 :
  600 DEF PROCmenu(x%,y%)
  610 $bk%="Formatter":bk%!12=&70207
  620 bk%!16=150:bk%!20=40:bk%!24=0
  630 bk%!28=0:bk%!32=-1
  640 bk%!36=&07090011:$(bk%+40)="Quit"
  650 bk%!52=&80:bk%!56=-1
  660 bk%!60=&07090011:IF mode%=-1 THEN
bk%!60=(bk%!60 OR &400000)
  670 $(bk%+64)="Abort"
  680 SYS "Wimp_CreateMenu",,bk%,x%,y%
  690 ENDPROC
  700 :
  710 DEF PROCgo(action%)
  720 LOCAL answer%,sector%,ptr%,size%,i
%
  730 SYS"Wimp_WhichIcon",handle%,bk%,&3
F0000,&220000
  740 IF !bk%=-1 THEN ENDPROC
  750 FOR i%=0 TO floppies%
  760 IF icon_drive%(i%)=!bk% drive%=i%
  770 NEXT i%
  780 SYS"Wimp_WhichIcon",handle%,bk%,&3
F0000,&210000
  790 IF !bk%=-1 THEN ENDPROC
  800 CASE !bk% OF
  810 WHEN icon_l%:format$="L"
  820 WHEN icon_d%:format$="D"
  830 WHEN icon_e%:format$="E"
  840 ENDCASE
  850 rec%!12=&203:rec%!20=RND
  860 $(rec%+22)=STRING$(42,CHR$(0))
  870 $(rec%+22)=$title%
  880 CASE format$ OF
  890 WHEN "L":
  900 rec%!0=&02011008:rec%!4=0
  910 rec%!8=0:rec%!16=655360
  920 WHEN "D":
  930 rec%!0=&0202050A:rec%!4=0
  940 rec%!8=0:rec%!16=819200
  950 WHEN "E":
  960 rec%!0=&0202050A:rec%!4=&1070F
  970 rec%!8=&05200100:rec%!16=819200
  980 ENDCASE
  990 defects%=0
 1000 IF action%=0 THEN
 1010 IF $title%="" THEN ENDPROC
 1020 !bk%=0
 1030 $(bk%+4)="Format "+STR$(drive%)+"
"+format$+" - Are you sure ?"
 1040 SYS"Wimp_ReportError",bk%,&17 TO ,
answer%
 1050 IF answer%=1 THEN
 1060 size%=rec%?0-7:ptr%=data%
 1070 PROCbytes(60,&4E)
 1080 FOR sector%=0 TO rec%?1-1
 1090 PROCbytes(12,0):PROCbytes(3,&F5)
 1100 PROCbytes(1,&FE):PROCbytes(1,0)
 1110 PROCbytes(1,0)
 1120 PROCbytes(1,sector%)
 1130 PROCbytes(1,size%)
 1140 PROCbytes(1,&F7):PROCbytes(22,&4E)
 1150 PROCbytes(12,0):PROCbytes(3,&F5)
 1160 PROCbytes(1,&FB)
 1170 PROCbytes(128<<size%,&E5)
 1180 PROCbytes(1,&F7):PROCbytes(24,&4E)
 1190 NEXT sector%
```

```
 1200 PROCbytes(1050-50*size%,&4E)
 1210 mode%=0:verify%=FALSE
 1220 ENDIF
 1230 ELSE
 1240 mode%=80:verify%=TRUE
 1250 ENDIF:ENDPROC
 1260 :
 1270 DEF PROCbytes(n%,byte%)
 1280 LOCAL i%
 1290 FOR i%=0 TO n%-1
 1300 ptr%?i%=byte%
 1310 NEXT i%:ptr%+=n%:ENDPROC
 1320 :
 1330 DEF PROCdo_track(drive%,format$,RE
TURN mode%)
 1340 LOCAL track%,action%,sector%,ptr%,
size%,fsize%,a1%,a2%,f%,errorp%
 1350 track%=mode% MOD 80
 1360 action%=mode% DIV 80
 1370 IF action%=2 THEN
 1380 PROCcreate_struc(drive%,format$)
 1390 mode%=-1:PROCupdate_bar(0,79,0)
 1400 SYS"XOS_CLI","ADFS:Dismount :"+STR
$~drive%
 1410 ELSE
 1420 CASE format$ OF
 1430 WHEN "L":
 1440 a1%=track%*4096
 1450 a2%=track%*4096+327680
 1460 WHEN "D","E":
 1470 a1%=track%*10240
 1480 a2%=track%*10240+5120
 1490 ENDCASE
 1500 a1%=a1% OR (drive%<<29)
 1510 a2%=a2% OR (drive%<<29)
 1520 CASE action% OF
 1530 WHEN 0:
 1540 size%=84+(1<<(rec%?0))
 1550 FOR sector%=0 TO rec%?1-1
 1560 ptr%=data%+sector%*size%
 1570 ptr%?76=track%:ptr%?77=0
 1580 NEXT sector%
 1590 SYS"ADFS_DiscOp",,4 OR (rec%<<6),a
1%,data%
 1600 FOR sector%=0 TO rec%?1-1
 1610 ptr%=data%+sector%*size%:ptr%?77=1
 1620 NEXT sector%
 1630 SYS"ADFS_DiscOp",,4 OR (rec%<<6),a
2%,data%
 1640 WHEN 1:
 1650 size%=rec%?1<<rec%?0
 1660 SYS"XADFS_DiscOp",,1 OR (rec%<<6),
a1%,data%,size% TO errorp%;f%
 1670 IF (f% AND 1) THEN PROCbad(a1%,for
mat$,errorp%)
 1680 SYS"XADFS_DiscOp",,1 OR (rec%<<6),
a2%,data%,size% TO errorp%;f%
 1690 IF (f% AND 1) THEN PROCbad(a2%,for
mat$,errorp%)
 1700 ENDCASE
 1710 PROCupdate_bar(track%,track%,11-ac
tion%):mode%+=1
 1720 ENDIF:ENDPROC
 1730 :
 1740 DEF PROCbad(addr%,format$,errorp%)
 1750 LOCAL i%,f%,size%
 1760 IF (!errorp% AND &FFFFFF) <> &108C
7 THEN ENDPROC
 1770 IF format$="E" AND NOT verify% THE
N
 1780 FOR i%=0 TO 4
 1790 SYS"XADFS_DiscOp",,1 OR (rec%<<6),
addr%+i%*1024,data%,1024 TO ;f%
 1800 IF (f% AND 1) THEN PROCmark(addr%+
i%*1024)
 1810 NEXT i%
 1820 ELSE
 1830 OSCLI"Error 0 Bad Sector at &"+STR
$~(addr%)
 1840 ENDIF:ENDPROC
 1850 :
 1860 DEF PROCmark(addr%)
 1870 IF defects%>254 THEN
 1880 *Error 0 Too many bad sectors
 1890 ELSE
 1900 dlist%!(defects%*4)=addr%
 1910 defects%+=1:ENDPROC
 1920 :
 1930 DEF PROCerror(err%,err$)
 1940 bk%!0=err%:$(bk%+4)=err$+" (Error
Code "+STR$err%+")"
 1950 SYS"Wimp_ReportError",bk%,&11
 1960 mode%=160:ENDPROC
 1970 :
 1980 DEF PROCcreate_struc(drive%,format
$)
 1990 LOCAL flags%,size%,ssize%
 2000 IF verify% THEN ENDPROC
 2010 CASE format$ OF
 2020 WHEN "L":flags%=&C0
 2030 WHEN "D":flags%=&40
 2040 WHEN "E":flags%=0
 2050 ENDCASE
 2060 dlist%!(defects%*4)=&20000000
 2070 SYS"FileCore_FloppyStructure",data
%,rec%,flags%,dlist% TO ,,,size%
 2080 ssize%=1<<rec%?0
 2090 size%=(size%+ssize%-1)DIVssize%*ss
ize%
 2100 SYS"ADFS_DiscOp",,2 OR rec%<<6,dri
```

We offer a unique inspection copy service

NOW RISC OS COMPATIBLE

ARCHWAY

Have you
spent long hours struggling
with the programming complexities of the brilliant
Archimedes **WIMPS** system?

YOU can easily and quickly build multi-window programs for Archimedes with pop-up menus, icons, mouse control, etc. of a professional quality. All in BBC Basic. With RISC OS your programs are multi-tasking and installed on the icon bar.

ARCHWAY provides a total environment for programming with **tools, run-time** code, program **shells**, extensive **examples** and a **library** of functions and procedures.

The **tools** let you define and edit windows, multi-level menus, icons, mouse pointers and much more. The cut, copy and paste, undo, on-line help and hard copy print-out facilities give you power as well as ease of use and convenience.

The **run-time** code becomes part of any program you create. It automatically handles windows, menus, icons, etc. for you. For example, opening a window can involve little more than calling a function with the window's name. Attach a menu name to a window using the tools and the menu functions without you writing a line of code. Give a menu item a command name and when a user chooses it a function in your code with the same name is called. You just write the contents of the function.

An in-depth user guide takes you gently step-by-step through a progressive series of program building examples. All examples are on disc ready to run.

£79.95
incl. VAT & p/p

The complete system comes on 4*800k discs (3 tools & 1 run-time) with a 300+ page, ring-bound user guide. You need 1M of RAM to run the tools. One disc drive is sufficient. No extra licence fee to distribute copies of the run-time in your programs.

AVAILABLE SOON: A Major New Release (Vl. 1). Many new features; eg powerful graphics and text facilities for your programs by multitasking cooperation with the RISC OS Draw, Edit and Paint programs. **BUY ARCHWAY NOW AND UPGRADE WITHOUT FURTHER CHARGE!**

SIMTRON *Programs to help you*

4 CLARENCE DRIVE, EAST GRINSTEAD,
WEST SUSSEX RH19 4RZ Telephone (0342) 328188

Access

Cheques/POs/official orders or Access/Visa number and expiry date. 24-hour 'phone for Credit Card orders.

VISA

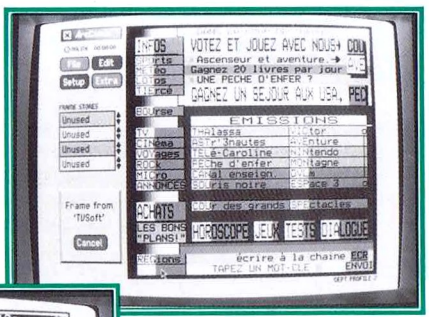
```
ve%<<29,data%,size%:ENDPROC
 2110 :
 2120 DEF PROCupdate_bar(start%,end%,col
our%)
 2130 LOCAL i%,j%
 2140 FOR i%=start%*3 TO end%*3+2
 2150 FOR j%=0 TO 4
 2160 SYS"OS_SpriteOp",&12A,spArea%,"Bar
Graph",i%,j%,colour%
 2170 NEXT j%,i%
 2180 SYS"Wimp_ForceRedraw",handle%,8+st
art%*6,-324,14+end%*6,-308:ENDPROC
 2190 :
 2200 DEF FNquit_ok
 2210 LOCAL answer%,quit_ok%,action$
 2220 IF mode%<80 THEN action$="Format"
ELSE action$="Verify"
 2230 IF mode%>-1 THEN
 2240 !bk%=0:$(bk%+4)="Abort "+action$+"
 - Are you sure ?"
 2250 SYS"Wimp_ReportError",bk%,&17 TO ,
answer%:quit_ok%=(answer%=1)
 2260 ELSE
 2270 quit_ok%=TRUE
 2280 ENDIF
 2290 =quit_ok%
 2300 :
 2310 DEF PROCclose_down(quit_ok%)
 2320 IF quit_ok% THEN
 2330 !bk%=handle%
 2340 SYS "Wimp_CloseWindow",,bk%
 2350 mode%=-1:quit%=TRUE
 2360 ENDIF:ENDPROC
 2370 :
 2380 DEF PROCwindow
 2390 LOCAL temp%,flags%
 2400 bk%!0=400:bk%!4=400:bk%!8=900
 2410 bk%!12=736:bk%!16=0:bk%!20=0
 2420 bk%!24=-1:bk%!28=&87030012
 2430 bk%!32=&01070207:bk%!36=&000C0103
 2440 bk%!40=0:bk%!44=-336:bk%!48=500
 2450 bk%!52=0:bk%!56=&3D:bk%!60=&3000
 2460 bk%!64=spArea%:bk%!68=0
 2470 $(bk%+72)="Formatter":bk%!84=0
 2480 SYS"Wimp_CreateWindow",,bk% TO han
dle%
 2490 temp%=FNicon(8,-288,102,-240,&C700
0011,0,0,0,"Name:")
 2500 temp%=FNicon(354,-56,462,-8,&C7000
011,0,0,0,"Drive:")
 2510 temp%=FNicon(8,-56,140,-8,&C700001
1,0,0,0,"Format:")
 2520 $(buffer%)="640k, Old Map"
 2530 temp%=FNicon(76,-112,296,-64,&C700
0119,buffer%,-1,14,"")
 2540 $(buffer%+14)="800k, Old Map"
 2550 temp%=FNicon(76,-168,296,-120,&C70
00119,buffer%+14,-1,14,"")
 2560 $(buffer%+28)="800k, New Map"
 2570 temp%=FNicon(76,-224,296,-176,&C70
00119,buffer%+28,-1,14,"")
 2580 icon_1%=FNicon(8,-112,66,-64,&C701
503D,0,0,0,"L")
 2590 icon_d%=FNicon(8,-168,66,-120,&C70
1503D,0,0,0,"D")
 2600 icon_e%=FNicon(8,-224,66,-176,&C72
1503D,0,0,0,"E")
 2610 SYS"ADFS_Drives" TO ,floppies%
 2620 icon_drive%(0)=FNicon(354,-112,408
,-64,&C722503D,0,0,0,"0")
 2630 icon_drive%(1)=FNicon(432,-112,486
,-64,FNff(1),0,0,0,"1")
 2640 icon_drive%(2)=FNicon(354,-168,408
,-120,FNff(2),0,0,0,"2")
 2650 icon_drive%(3)=FNicon(432,-168,486
,-120,FNff(3),0,0,0,"3")
 2660 title%=buffer%+42
 2670 $(title%)=""
 2680 $(buffer%+53)="a~ "
 2690 icon_name%=FNicon(110,-288,296,-24
0,&0700F13D,title%,buffer%+53,11,"")
 2700 icon_format%=FNicon(354,-228,488,-
176,&0B03903D,0,0,0,"Format")
 2710 icon_verify%=FNicon(354,-288,488,-
236,&0D03903D,0,0,0,"Verify")
 2720 icon_bargraph%=FNicon(8,-328,488,-
304,&07000006,0,0,0,"BarGraph")
 2730 !bk%=handle%
 2740 SYS"Wimp_GetWindowState",,bk%
 2750 SYS"Wimp_OpenWindow",,bk%
 2760 ENDPROC
 2770 :
 2780 DEF FNff(drive%)
 2790 IF drive%>=floppies% THEN = &C7425
03D ELSE =&C702503D
 2800 :
 2810 DEF FNicon(x0%,y0%,x1%,y1%,flags%,
data1%,data2%,data3%,text$)
 2820 LOCAL han%
 2830 bk%!0=handle%:bk%!4=x0%:bk%!8=y0%
 2840 bk%!12=x1%:bk%!16=y1%
 2850 bk%!20=flags%
 2860 IF text$="" THEN
 2870 bk%!24=data1%:bk%!28=data2%
 2880 bk%!32=data3%
 2890 ELSE
 2900 $(bk%+24)=LEFT$(text$,11)
 2910 ENDIF
 2920 SYS"Wimp_CreateIcon",,bk% TO han%
 2930 =han%
```

ArcComm Review

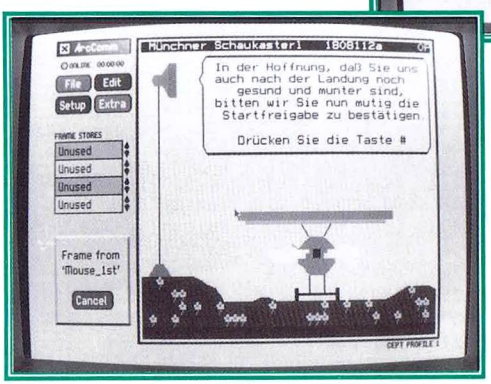
Peter Rochford looks at the latest comms package for the Archimedes.

ArcComm has been a long time coming as they say. This new comms package for the Archimedes from BBCSoft has been rumoured to be ready for release for the past six months or so. It was written by Peter Gaunt, the author of the Weirdbeard and Modem Master comms software for the Beeb.

The *ArcComm* package comes supplied in a video cassette-style case and comprises two discs, a 120-page spiral bound manual and a function keystrip. A thoughtful touch is the inclusion of a printed disc label to use when you make a working copy of the main disc.



Teletel



Bildschirmtex

ArcComm is designed to work only with RISC OS, and runs under the Desktop. However, sadly it does not allow true multitasking, as you have to close down the ArcComm window totally to load a new application. This leaves the ArcComm package installed, and if online it will hold the communications link whilst you do something else. Don't get too excited about this though, as on a 1Mbyte machine you have precious little memory left to run the majority of applications.

The window environment in *ArcComm* is not standard RISC OS. The author has opted for his own which, although I think it is very attractive, is somewhat slower. Don't let that worry you too much though, as in general use, it does not make that much of a difference. However, I do feel that developers should try and adhere to the guidelines for appearance set down by Acorn. After all, they are designed to make life easier for the user.

The package features seven types of terminal emulation all of which are selected from drop-down menus. Three of these are viewdata or videotex, namely CEPT Profile 1, CEPT Profile 2 and CEPT Profile 3. The rest are the four most common scrolling text terminals, Teletype, VT52, VT100 and ANSI.

A word of explanation about the videotex terminals will be needed to clarify what CEPT Profiles are. CEPT Profile 1 is the system used by the German Bildschirmtex videotex system, CEPT Profile 2 is used by the French Teletel system and CEPT Profile 3 is the British Prestel-type viewdata system. A full explanation of the whys and wherefores of CEPT is not really within the scope of this review and somewhat of a digression. The second disc supplied with ArcComm features sample screens from all three systems. If you have never seen the German Bildschirmtex system then you are in for a shock. It features high res graphics in up to 32 colours. Prestel by comparison looks downright crude. Wake up BT!

Back to *ArcComm* itself. The package offers three options for downloading data: ASCII, Xmodem and CET for telesoftware. The usual buffering, spooling and printing facilities are provided along with the saving of single frames or a number of frames to any one of four frame stores for later recall.

The videotex terminals have the ability to edit offline mailbox frames for you to upload when on line. This uses templates to represent the standard mailbox frames. These have fields defined on them exactly the same as their on-line equivalents. This means that they operate exactly the same as if you were typing them in on line. Not only that, but a character count is kept that also takes account of Escape codes so that you cannot enter more characters than the online frame would allow. This makes offline editing child's play.

You can use *ArcComm* with a dumb modem or a Hayes modem, the only other driver included is for the Dacom, as supplied by the DTI to schools. No facility exists within the package to create your own as in *Hearsay* or indeed modify the existing ones. It is possible to write your own and load it as a relocatable module, but I cannot see too many people doing this.

There is no telephone directory within *ArcComm* for you to set up to allow selection of a database to dial up. Instead, there is a limited form of script language similar, but nowhere near as powerful as the one in *ArcTerm*. This allows you to create procedures using a number of keywords to set up a sequence of tasks for *ArcComm* to perform to effect the logging on to a particular dialup service. Once created, these are selected by clicking on them from the Desktop, and they will invoke *ArcComm* automatically, set the correct terminal and baud rate, dial up the number and send your password and ID.

Most of the operations within *ArcComm* can be performed using the mouse via the drop down menus or selecting one of the small number of displayed icons. In the videotex terminals, selection of routes can be done either by clicking on the displayed menu or clicking on a keypad alongside the main screen. An alternative method of selecting terminal features is by use of the functions keys, or you can use the main keyboard in combination with the Action (Ctrl) key.

Documentation supplied with *ArcComm* is unreservedly good. The manual is a rather unusual size, being wide but not very deep. It is wire bound, but in such a way that you can read the spine when the book is placed on a shelf. The layout is very clear with large left margins for sub-headings, and crisp diagrams. There is a substantial index and an excellent reference section, which includes modem wiring details. As well as covering all aspects of *ArcComm* from a user's point of view, the manual also deals with writing your own modem driver.

CONCLUSIONS

When you consider the price of *ArcComm,* it really is hard to fault it. It is professionally finished and packaged, and is a pleasure to use. Comparisons to the other two major Archimedes comms packages must obviously be made. *ArcTerm* and *Hearsay* are both much more powerful pieces of comms software and have a wealth of extra facilities over *ArcComm,* but then they are twice the price.

ArcComm scores as being a straightforward comms setup for those who need all the basic viewdata and scrolling text terminal features in an easy-to-use package. Nice as they are to have, the inclusion of the two European videotex terminals is a questionable decision as few people I doubt will ever use them.

For my own use *ArcComm* would not suffice as it lacks the power, flexibility and facilities of Hearsay, which I use extensively. Having said that, there are certain features of *ArcComm* that I would dearly love to see in *Hearsay,* namely the excellent mailbox editor and some form of script language.

For many people though, *ArcComm* would be a good choice of comms software and if your needs do not extend to those provided by the likes of *ArcTerm* and *Hearsay,* then I would wholeheartedly recommend it.

Product	**ArcComm**
Supplier	**BBCSoft, P.O. Box 234, Wetherby, West Yorkshire, LS23 7EU.**
Price	**£29.95 inc. VAT + £1.50 p&p**

RU

ANNOUNCING A MAJOR NEW SOFTWARE RELEASE

Premier

Circle Software are proud to announce a major new software release for the Archimedes. Premier is so new, there is as yet no defined category for it. Is it a word processor? Is it a report generator? Is it a spreadsheet or a data base? Premier fits none of these categories, rather it spans them all. Premier is a major advance in capability from a single program. These are just some of its features -

The ability to read and execute commands from files allows any text processing to be carried out just by typing a 'star' command, or picking from the desk-top. Further, the ability to define 'macro' commands, and to assign these to ANY key, allows complete 'applications' to be built, including customised spreadsheets, and simple data bases. A command file defining a full featured word processor, with matching key-strip, is included with Premier.

This word processor alone justifies the purchase of Premier. It allows text to be moved between any number of files open at once, searches to be made forwards or backwards using powerful pattern matching, document headers and footers to be defined, has on screen formatting and other features too numerous to list here. The command file defining this may be edited to tailor the features to your own requirements, a feature not found in any other word processor.

DYNAMIC TEXT is a new concept which allows the inclusion of variable items within any document. Such items may show today's date or time, the current value of VAT, or a name and address. The actual text displayed may be computed, defined at the keyboard, or read from a data file. This system facilitates the production of statements and invoices, financial reports, or personalised mail shots.

Premier's powerful command language, which includes maths and string functions, repeat loops and 'if - then' constructs, may be included in the definition of a DYNAMIC TEXT item, allowing full spread-sheet type facilities anywhere on the page, not just in neat rows and columns. This also allows printer codes to be sent, providing the simplest method yet devised to control printer functions from embeded commands.

Now fully RISC-OS Multi-tasking.

Write or phone NOW for full details, of Premier, a £5 refundable demo disc or the full version of Premier.

Circle Software

33, Restrop View,
Purton, Swindon,
Wilts. SN5 9DG
Tel: 0793 770021

Access

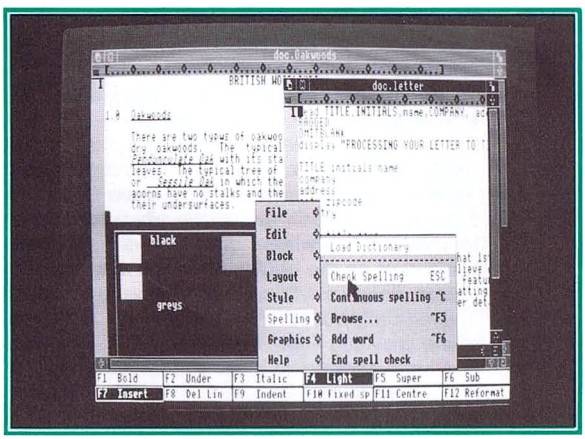
Price including 160 page manual and applications, £96 + VAT, post free.

VISA

Making the most of 1st Word Plus

Mark Sealey recounts his experiences with 1st Word Plus, and offers some helpful tips for other users.

Like it or loathe it, 1st Word Plus is the most substantial word processor available for the Archimedes. Many loved it because it came to our rescue (over a year ago) on the heels of the ill-fated Arcwriter. 1st Word Plus is not an entirely in-house product. Originally written by GST for the ST range, it nevertheless operates very well in the Archimedes WIMP world: a header to allow it to be started from the RISC OS Desktop is on the Acorn RISC OS support disc, though 1st Word Plus will not multi-task. In other respects it is RISC OS compatible.



Editing two documents with 1st Word Plus

What follows is an indication (based on eighteen months of almost daily use) of how to get 1st Word Plus to do the routine tasks as well as some of the more advanced ones expected of more expensive word processors. What is more, it will concentrate on those features which are either likely to cause the most trouble or save the most time. The topics touched on fall into five main categories: files, keying, printing, spelling and mailmerging.

SHORTCOMINGS

Some of this implies that despite several upgrades (Acorn will update versions earlier than that of June 1988 if returned to them) it is still less than perfect. Incredibly, 1st Word Plus, has neither case-swap nor access to the operating system - for example to unlock files. It has a directory structure complicated enough to deter anyone unfamiliar with ADFS, and can behave unpredictably when changing text styles on screen and during formatting. There is no *key* combination to move to the top or bottom of files. At times, text jumps to the right hand side of the page and doesn't reappear on exit from the hyphenation dialogue box.

Acorn is aware of most of these drawbacks and will, presumably, put them right if and when a fully updated version is released. 1st Word Plus nevertheless supports several powerful features such as the incorporation of graphics into documents, a good hyphenating routine, and footnotes. It is also easy to work on more than one document on screen at once.

DISCS AND FILING

In the first place there is an excellent 250 page manual. It takes you through the basics of text entry, editing, formatting and filing. On booting, 1st Word Plus requests a filename for editing, in a window which will become extremely familiar. By default, this file is in the DOC directory and is named "untitled". If you want to create a new file you enter your text first and name it when it is saved. Otherwise, if you wish to load an old file, simply click on its name or change directories to find it.

If the file you create isn't in the DOC directory (see below), be sure to convert it to 'WP mode' by clicking on the relevant box in the Edit sub-menu. Only in WP mode can all the operations which you need for formatting, blocks, find/replace and multiple rulers be carried out. And only when it has been saved can you print a file. That is the time for final options for printer margins and number of copies etc. to be set.

The windows containing filenames only appear one directory at a time, but you can cycle up and down any directory structure very easily by clicking on the cross symbol normally associated with closing a window. If you are in one directory and the file you want is in another, you need to follow this procedure. The file's name must actually appear in the filenames list on screen. This is particularly relevant if there is a document which you know exists but is not in the currently selected directory. This often happens when you come to print out or to read a file from another directory into the document being edited. What is more, if 1st Word Plus senses that the text has been changed, it has to be saved again before it can be printed.

When you save a file (say *letter5*) with a name that already exists, it re-names the earlier version to *BAK.letter5* or whatever. You may wish to create a sub-directory called LETTERS if the DOC directory gets too full for all the filenames to appear in the window at once. If so, use *CDIR from outside 1st Word Plus to create both LETTERS and BAK on the same directory level, that is below DOC in the hierarchy.

If you have a double disc drive, 1st Word Plus can be made to default to drive 1 for DOC (or indeed any other) files. Working from your backup copy of the 1st Word Plus distribution disc, alter the line (line 30 in most versions) which sets the system variable *1stWordPlus$Docs* in the Basic program $.Library.1stWord+ from "$.1wp.doc" to ":1.$.1wp.doc" - case is not important, of course.

KEYING CONVENTIONS AND USAGE

One of the strengths of 1st Word Plus is that what you see is almost exactly what you get. Width of print (condensed, enlarged etc) cannot be seen on screen, though. Since these effects are brought about by changing the ruler, the only way to know which style is selected is to keep an eye on the ruler as it changes whilst you scan through your text. However, text effects such as italic, bold and super/subscript, as well as non-standard font characters are actually visible on screen. But the exact procedure for obtaining these is unclear from the manual.

Here is the sequence which effectively marks out small (or longer) blocks: click with the mouse pointer on the first character you want highlighted, drag the now altered pointer along and if necessary down to the last character to be highlighted. Should some of the text be outside the screen window, Ctrl-S and Ctrl-E may be used instead with the cursor keys to mark the start and end. The letter(s), word(s) or line(s) of text selected are partially blocked out. Next press one or more of function keys f1 to f6 according to the effect desired. They act as a toggle and can be combined. The text - still partially obscured by the blocking - will change to italic, underline or whatever. To complete the effect, either reformat (f12) or begin to mark another block. To de-select the highlighting altogether, click with Adjust anywhere on the block.

At times, text will jump up - as it does with View - onto the previous line; even when you have formatted it - especially after deleting an intervening line with f8. The answer is to place the cursor on the beginning of the wayward line of text, delete anything that may be to the left of it, eventually concatenating it up next to the end of the previous line. Re-inserting a Return there pushes it back down once and for all.

PRINTING

You might like to consider attaching Lingenuity's Control Panel in the 1st Word Plus !BOOT file so that it always starts up with an appropriate machine configuration. And while you're at it, what about the RISC User Printer Buffer utility (see Volume 1 Issue 3)? I find 64K is a good size for the buffer. Again the place to install it is in that same Basic program referred to already. The lines needed to do this are:

```
*RMLOAD BUFMODULE
*BUFFER ON
```

An area about which much has been written since 1st Word Plus appeared is printer drivers. Assuming you have managed to translate the printer sequences into a usable table, these will be saved in the $.1WP.cfg file.

Nowhere has it been made explicit how to change the default driver. First edit the appropriate file in the $.1WP directory as described in the 1st Word Plus manual - Appendix B. Next copy the 'cfg' file produced by running the Printer install program to the *$.Resources.1wp* directory. You must search your 1st Word Plus working disc for the file "1wp_Print" in the directory *$.Resources.1wp.*

Delete the file (or re-name it - e.g. to OLDprint - if you wish to preserve it in case of errors). Then re-name to 1WP_Print either the file you have been working on or one already provided to match your printer. Do not be put off by all this; one of the strengths of 1st Word Plus is the happy combination of Basic and Absolute code to maximise both user-accessibility and speed.

SPELLING AND MAILING

These are two final examples of such strong points. Spelling checks take place at a truly amazing rate, and are easy to perform. Once the dictionary is loaded (via the Spelling menu) pressing Ctrl-C will check as you type, although don't be surprised when hyphenated words are not recognized.

Even greater power comes from combining dictionaries, and an excellent suite of three has just been published for £6.95 inclusive by Science Frontiers. You can merge everything into one giant, combined specialised and standard dictionary using the 1stDmerge utility outside 1st Word Plus. This is the only way to add new words to the main dictionary. Merge the main one with the 'supplementary' file into which additions are all saved by default. Even then there will still be room for all the systems and utility programs on the same disc - though not really for data and text files as well.

As for mail shots, 1stMail (which has a long and extremely clear tutorial section occupying a quarter of the manual) is also loaded separately from 1st Word Plus. It obeys the same conventions in a familiar environment. There is one drawback though: page breaks do not work consistently. To get round this, spool the merged file to disc for reloading into 1st Word Plus rather than send it directly to the printer. This means cycling through the output box at the bottom right of the screen first.

Macros are fully implemented, drawing either on direct keyboard input or a data file (which must be in the correct, DAT, directory). Minerva's Mailshot (an excellent product in its own right) is ideal for creating such files in a format that 1st Word Plus (and indeed most other word processors) can use. 1stMail can also handle whole files as macros so that a letter could effectively call a standard letterhead or paragraph(s) which you regularly wanted to be included. These can be nested, and their use is well explained in the manual.

If the absence of certain standard features mentioned earlier is the sorest point with 1st Word Plus users, the flexibility and user-friendliness of 1stMail are among the package's strongest points. With some similar products, the combination of mouse and keyboard is counter-intuitive and even intrusive. Whatever you've heard about 1st Word Plus, give it a try.

WORD PROCESSING/DESKTOP PUBLISHING

Those who feel that a word processor such as 1st Word Plus does not offer all the facilities for word processing and page layout that they would wish, might want to wait for one of the new breed of word processor/desktop publishing packages expected very shortly. Acorn Computers, Computer Concepts and BEEBUG all have such products at an advanced state of completion, and all three are likely to be demonstrating their latest software at the BBC Acorn User Show in July.

SUPPLIERS

Minerva
69 Sidwell Street,
Exeter EX4 6PH.
Tel. (0392) 437756

Science Frontiers
7 Porthill Court,
Aberdeen
AB1 1DU.
No phone

Lingenuity
P.O. Box 10, Halesworth
IP19 0DX.
Tel. (098) 685476

RU

Dabhand User Update

Archimedes Basic Compiler Version 2 *the* Quality BASIC V Compiler
OS Guide · BASIC V Guide · Shareware Volume 2 · Software Special Offers

a rchimedes
b asic
c ompiler

Version 2 Now Available!

ABC is now being used commercially by software houses around the country to develop their own products. Repton 3 and Presenter 2 are just two examples of what can be done with ABC.

Here's what the computer press said about ABC version 1:

"*...Excellent Dabs Press product. Buy it!*" RISC User

"*...I can tell you now, I am very impressed. This is a superb package.*" Archive

"*ABC is a vital part of any programmer's toolbox, it puts compilers on other systems to shame. Unquestionably one of the most impressive pieces of software I have yet seen running on the Archimedes.*" A&B Computing

ABC2 offers even more including the following new features:

RETURN Parameters
Multiple exit PROCS/FNs etc
Double and Extended precision fp with no loss of speed
Shared runtime libraries
Global TYPEing
Remove FP instructions
New version 2 manual
Technical support and backup
Local Error Handling
RISC OS Desktop compatible (Installs on Icon bar)
Further extended CALL and USR
Automatic Register labelling
Optional Command Line operation
Improved HEAP and STACK directives
Extra Examples
Still no royalties to pay on compiled code

Coming Soon: Utilities disc with Profiler, Cross referencer and Library maker plus ABC65 the BBC 6502 code generator. Each supplied with comprehensive manual.

FREE DEMO DISC: To receive your free ABC demo disc which includes a cut-down version of ABC simply send us a sae and we'll do the rest! Benchmarks and specification on request. *ABC1 users you should have now received version 2 automatically, free of charge.*

£99.95

Archimedes OS Guide

The price of the 320 page guide is just **£14.95** or **£21.95** including the programs disc (with over 40 programs) and manual.

BASIC V: Dabhand Guide

For anybody interested in BBC BASIC then this book is essential reading. Assuming a familiarity with BBC BASIC the various many new components of BASIC V are fully described using example programs throughout.

An essential aid for all Archimedes owners, and including coverage of RISC OS BASIC. Price **£9.95** - 128 pages - available now.

SPECIAL OFFERS!

PC Emulator FREE Shareware

Purchase the Archimedes PC Emulator from us and we'll *give* you ARC-PC Shareware pack worth £34.95 Price just **£113.85** inc.

ANSI C & FREE C Dabhand Guide

The best combination! Purchase Acornsofts ANSI C and we'll throw in our top-selling Dabhand Guide to C - 512 pages and worth £14.95. Price just **£113.85** inc VAT & p&p.

Archimedes Games!

Arcendium: Board Game Fun

Backgammon, Draughts, Reversi and Quadline - four games for the price of one! **Just £14.95.**

Alerion: Arcade Action!

The highly acclaimed all-action shoot-em-up for the Archimedes. Impossible to finish! Superb fun! **Price £14.95.**

See us at the Acorn User Show - Stand 98.

New!

PC Emulator Shareware Volume 2

Now available is Volume 2 of the Shareware Collection. Four Megabytes of PC software on five discs, all tested with the emulator to ensure compatibility. Volume 2 includes a comprehensive database, text editor, typing tutor, astronomy package and much more. Price £34.95.

Shareware Volume 1

Five discs as above, the collection includes software you would normally expect to pay a fortune for. Volume 1 includes a wordprocessor, spreadsheet, games, flowchart designer, printer utilities, and more. Price £34.95.

DABS PRESS

5 Victoria Lane (RUJ), Whitefield, Manchester M25 6AL

Tel: 061-766 8423 BT Gold: 72:MAG11596 Prestel: 942876210

Prices include VAT and P&P (UK/BFPO/CI). ACCESS/VISA/STYLE accepted by post/phone/Mailbox/ in person. Cheques and POs to address above. Dabs Press products available from all good dealers. Add £2.50 (£12 air) if outside UK. Official orders welcome.

WIN AN A3000 COLOUR SYSTEM

IN THE BEEBUG/RISC User SUMMER COMPETITION

Acorn has generously agreed to provide an A3000 colour system (worth over £999) as the main prize in our Summer Competition.

In 1987 Acorn donated an Archimedes A305 colour system as a prize for our "Alphametics" programming competition. The interest and response to this was so impressive that we propose another competition along similar lines, involving the submission of a program which will be tested for speed and versatility. The competition will be open to all members of BEEBUG and RISC User. All programs will be timed on the same machine, thus making it fair for all. All programs must be written in a subset of BBC Basic which will run on any Acorn machine (except the Atom) without the use of any machine call, indirection operators, or operating system calls. This means that Master owners cannot use ON-PROC, while Archimedes owners must also avoid WHILE, CASE, IF-ENDIF, matrix operations, local error handling, RETURN parameters, local arrays, and the operators +=, -=, *=, /=. Programs may, however, use the standard Basic disc filing commands (DFS or ADFS) providing the file size does not exceed 100k.

WHAT YOU HAVE TO DO

Write a program to generate and print an expression using exactly four fours, which will produce the numbers 0 to 50 together with the expressions that generate them. These four fours may be combined in any way, provided that fours are the only digits that appear in the expression. Mathematical symbols are + - * / ^ ( ) . ! R G (R and G represent "Sqr Root" and "Gamma" respectively).

Here are some examples:

```
0=4+4-4-4        1=44/44
12=4/.4+4/R4     17=4*(4+R(R4^(-4)))
20=G4-G4+4!-4      48 =!4+!4+4-4
49=(R4+R4/.4)^R4
```

Where R means "Square Root", ! means "Factorial, and G means "Gamma".

```
  !n=n*(n-1)*(n-2)*...*1 e.g.   !4=4*3*2*1=24.
Gn=!(n-1) e.g. G4=!(4-1)=!3=6.
```

The examples above do not necessarily give the simplest solution. E.g. 20 could be represented more simply as 4*(4+4/4). Your program must be able to be easily changed to work with numbers other than "fours", for instance it should allow four "threes", or four "fives".

Send a postcard to be received by 31st August 1989 containing your ***Name, Membership Number, Day-time phone number, Time taken for 0-50, and the Machine used.***

Send to **BEEBUG A3000 Competition,**
117 Hatfield Road,
St Albans,
Herts, AL1 4JS
(please note that this is our new address as from 7th August).

If you are shortlisted, we will then ask you to submit your program, when it will be tested back to back against other programs. Do NOT be put off if you have a slow machine or if your program takes all day to provide expressions, or fails to find all solutions (they may not exist!) this will be taken into account in selecting the shortlist.

CRITERIA FOR JUDGING

Entries will be judged on a combination of the criteria listed below, and the winner must be able to supply a version suitably annotated for publication. BEEBUG will be the sole judge as to whether an entry conforms to the spirit of the competition, and sole judge when determining the winner. BEEBUG's decision will be final. No BEEBUG staff or their relatives are eligible to enter.

1. Speed - the program that finds expressions for the numbers 0 to 50 in the shortest time when run on the same machine as all the other entries.

2. Choosing simplest solutions - The program that gives the simplest expression for each number. Priorities for defining simplicity will be the following order + - * / ^ ( ) . R G "tricks", "approximations". For example the following solutions for the number 1 are graded in order of simplicity:

```
1 = 44/44
1 = (4*4)/(4*4)
1 = (4/.4)/(4/.4)
1 = G4/G(4-.4+.4)
1 = TAN(44+4/4) classed as a "trick"
1 = 4^(4/44!)  classed as an "approximation"
```

3. Digit Flexibility - The program should be simple to change to make it work for other digits such as four "threes".

4. Range Flexibility - The program should allow the range to be selected so as to find solutions from 50 - 100 say.

We are unable to undertake any correspondence on the competition, however we can supply a sheet expanding on various aspects upon receipt of an SAE.

TWO BASIC COMPILERS

Reviewed by Mark Sealey

INTRODUCTION

The first Basic compiler for the Archimedes (version 1 of ABC from Dabs Press) was reviewed in RISC User Volume 2 issue 2, and readers are referred to that issue for an introduction to the role of compilers in general. After a long wait, ABC2 (which will be supplied free of charge to all existing ABC owners) has been released together with a rival product, RiscBasic from Silicon Vision.

This comparative review looks at the operation of each package based on the following five criteria: features; ease and flexibility of use; speed of compilation and execution; size and manageability of object code; documentation and presentation.

In the interests of fairness, the two compilers have been tested using identical source programs (with half from Dabs and half from Silicon Vision and other common benchmarks) plus a representative selection of published and unpublished Basic programs, typical of what any Archimedes user is likely to have to hand.

RISCBASIC

RiscBasic will run under Arthur 1.2 or RISC OS, though the version available at the time of review was not multi-tasking. By the time you read this, a fully multi-tasking version of RISC OS, which I have seen in pre-release form, should be available. This does not alter the RiscBasic compiler itself, but the environment in which the compiler runs, and all the current features of the compiler will remain the same.

RiscBasic is then installed as an application on the menu bar, allowing Basic source files to be compiled by merely dragging the appropriate file icon onto the RiscBasic icon on the Desktop menu bar. However, it still appears to contain features, such as assuming directory pathnames, which are incompatible with Acorn's RISC OS philosophy. This version will be sent out to all purchasers of RiscBasic who have registered with Silicon Vision.

In use, the compiler can be invoked with a single command, which may contain parameters (marked with a minus sign), and must also contain the name of the source code to be compiled, e.g.:

```
RiscBasic -verbose Myprog
```

RiscBasic can also be menu driven. In this case the user is presented with an uncluttered and highly legible screen with two windows. The first gives access to operating system commands and allows the user to run the compiler. The other window permits the selection of up to 17 options, which fall into four broad groups affecting speed, comprehensiveness of compilation, the management of source and object code and error-checking. A **-debug** option, for instance, provides run-time error reporting. Clicking on any one of these options is exactly equivalent to adding it as a parameter if RiscBasic is called as a star command.

The **-turbo** option will produce faster code if FOR-NEXT and REPEAT-UNTIL loops have single exits. The **-width** option allows the inclusion of the Basic keywords WIDTH and COUNT but produces slower code. Two options **(-OS0** and **-nomap)** produce code compatible with earlier versions of the Operating System than 1.2 while **-single** forces single instead of double precision floating point arithmetic.

The **-verbose, -stacktest, -nowarnings** and **-spool** options assist in error-checking and tracking what is happening during compilation. Cross referencing is provided by **-dump.** There are also useful options to change the object filename and to drop you into the ARM Basic Editor at the errant line if a source error should occur. RISC OS no longer holds the Basic Editor in the Operating System ROMs so it must now be loaded first. However, on exiting from the editor, it is still necessary to restart the whole process. With both packages, this is easier and more elegant in the Desktop environment.

Basic programs can be compiled into relocatable modules, including ones with multiple star commands, and RiscBasic comes with many useful demonstration files. As with ABC1 and ABC2, directives (up to 27 here) can be placed in source programs to control some of the compiler options just described - as well as operations like conditional compilation, constant and register variable allocation, and relocatable module functioning.

Before looking at the performance of RiscBasic, one or two other features must be mentioned: rather than leaving sections of ARM Assembler code within a Basic program to be

assembled each time the object is executed, Silicon Vision's product assembles the source code and links it, placing it at the end of the compiled code, but this does rule out conditional assembly, and the assembled code can no longer directly access the values of any Basic variables, which is a major limitation.

Silicon Vision's quite plausible point of view is that a Basic compiler is a tool to be used quickly and without fuss. Details contained in their 20-odd page documentation are more than adequate. Run RiscBasic: everything in BBC Basic V (apart from EVAL and the keywords associated with HIMEM and INSTALL), will compile. Let the compiler do the work. Needless to say, the compiled code will multi-task where appropriate.

Silicon Vision has a commendable policy of continually updating its products. The addendum to the manual contains details of minor improvements, for example, in module handling. The buyer who has filled in the registration card may be assured, though, of getting major upgrades free of charge.

ABC2

ABC1 was already a substantial product, and purchasers of the original receive a free upgrade and manual. This is just 30 pages long and lists the improvements, fixes and changes, of which more later. As before, there is a separate examples disc, which contains the old programs as well as some new ones, with very honest notes as to what changes in speed are likely to result from compilation. An ABC 'extras' disc is expected to be released in the near future containing a cross-referencer, library-maker, and profiler and tracer.

The compiler (v.2.29) will now also start from the Desktop, and runs as a non multi-tasking application under RISC OS. Files to be compiled are simply dragged onto the ABC icon. Otherwise there is little difference in appearance or use. Because the library routines have been put into a module, this must be present for any object code to run.

As before, code is given default names and can be compiled from disc or memory. The cycle of compile-debug-compile can become quite mechanical, and it is unfortunate that the filename windows in the Arthur version neither retain the current name nor allow one-key deletion, and that the same default file name ('Object') is always given under RISC OS.

It can still be run either from the command line or from its menu screen. As before, there are one or two user-unfriendly features: the default file-paths suggested during installation (which only works for Arthur 1.2 anyway) do not work with a single disc drive. On the other hand, commands for allocating heap and stack space have been improved. Like its rival, the Dabs compiler is undergoing continual upgrading, and any major changes will be supplied to existing owners free of charge.

Some of the additional features supported by ABC2 are: double and extended precision (19 digits) floating point variables. These are explained most clearly in the documentation as is the use of the TYPE directive to control them. Local error handling has been vastly improved. So has implementation of procedure RETURN parameters. The rules relating to procedure structure have been relaxed. There are also improvements in a number of areas associated with register and variable names.

Some twenty odd known bugs have been fixed and these are well documented. Features still not supported (which are present in RiscBasic, though) are local arrays, multiple entry to FOR, REPEAT loops and some Basic keywords (COUNT and WIDTH, for instance) in addition to the usual limitations with HIMEM, INSTALL and the like.

COMPARATIVE TESTING

By far the fairest way of comparing the two compilers in any quantitative way is to take a variety of benchmarks (supplied equally by the two publishers), run them under identical conditions, and compare the results side by side (bearing in mind the limitations of benchmarks as such).

The following results emerged from averaging ten recorded timings, all in mode 0 with the background check for Escape enabled, using single precision floating point arithmetic so as most closely to resemble interpreted Basic V. All tests were run on a 410/1, loading both the compiler and source code from a floppy, not a hard disc.

To start with, the same benchmarks as were published in the earlier review were used with

both compilers. The time (in seconds) is that taken to RUN each routine.

	Basic V	ABC2	RiscBasic
GrafScreen	1.46	0.85	0.71
TextScreen	2.19	2.03	1.99
IntMath	1.60	0.29	0.15
RealMath	0.22	0.24	0.25
Triglog	1.00	2.76	2.73
Sieve	4.52	0.45	0.06
Fibonacci	7.33	1.15	0.14
Ackerman	4.01	0.27	0.16
IntArray	1.29	0.32	0.05
RealArray	1.38	1.15	0.09
StringArray	0.88	1.03	0.11

The following loop benchmarks (100,000 iterations) were also run on each version:

ForLoop	18.35	14.43	1.34
RepeatLoop	9.94	1.43	0.09
WhileLoop	7.68	1.42	0.11

A representative selection of the (mostly graphics) files supplied by DABS:

3D	1.02	1.01	1.01
Bounce	10.06	2.08	2.02
Fireworks	7.24	3.56	3.72

and lastly three more substantial programs:

Stringsort(1)	124.01	19.09	8.18
Sort (2)	3.77	0.36	0.13
Raytrace (3)	15.36	7.94	7.83

(1) 200 6-character strings - bubble sort
(2) 200 random integers in two arrays
(3) timings in minutes

The average resulting object code sizes and compilation times were similar. If anything RiscBasic's were better, despite having to contain all runtime routines (ABC uses an additional module). The two compilers were also well matched for accuracy.

Yet, such benchmarks must not be taken just at face value for a number of reasons.

Benchmarks such as Grafscreen are limited by the speed of the operating system and test its speed more than the number-crunching capabilities of a compiler. If all those benchmarks which do not rely heavily on the optimisation routines of the compiler (such as Sieve and Fibonacci) are excluded, the superiority of RiscBasic's speed is all the more striking.

In attempting to provide a fuller compiler, coping with all Basic V syntax, RiscBasic incurs certain necessary overheads which you would expect at times to slow it down. In fact - thanks to its more efficient library routines - this is not the case as is shown, for instance, by the bubble sort routine handling Basic's SWAP keyword.

This also explains why the timings are inconsistent in favouring first one then the other product, and makes the computation of meaningful averages so difficult. In the end, the choice is between the better of the two option routines within the compilers. RiscBasic's seems to have the edge.

The benchmarks nevertheless indicate that code compiled by RiscBasic does run faster than that by ABC2, and by nearly 60% on average. What's more, RiscBasic is over three times faster than interpreted Basic V.

In most cases RiscBasic also compiles more quickly than ABC2 (on a range of programs, compile times varied from 7sec to 59sec with ABC2, 4sec to 24sec using RiscBasic), but it does take some extra 12 seconds to reload the RiscBasic compiler each time. The compiled object code requires substantially more memory than the original Basic program, and RiscBasic is more demanding in this respect than ABC2. For example, the same selection of programs ranging from 3K to 10K in size, compiled to 8K to 21K with ABC2, but 23K to 37K with RiscBasic. However, remember that at run time, ABC loads a library module which wipes out its advantage here.

You should also be prepared for some disappointment, when trying to compile your own Basic programs, arising from the occurrence of syntax unacceptable to the compiler. In practice, using a random selection of RISC User programs, this happened quite frequently, though more often with ABC2 than with RiscBasic. One small bug was located in the RiscBasic compiler (since corrected by Silicon Vision), but it is possible there are more.

CONCLUSIONS

Both products are very well presented, and well supported by their respective publishers. In

Continued on page 59

BBC TO ARCHIMEDES SCREEN CONVERTER

Change Beeb screens into Archimedes screens with this utility from Dag Haakon Myrdal.

When you consider that the Archimedes offers all the screen modes provided by the original model B, one might naively think that a screen could be saved from the Beeb and loaded directly into the Archimedes. However, this is simply not possible because while the modes are compatible in terms of resolution and colour, the way the data is stored in screen memory is totally different.

The accompanying program overcomes this barrier by taking a Beeb screen (in the form of a file), and converting it directly into an Archimedes screen, which can then be saved if desired. To use the program, type in the listing and save it. When you run it, you are asked to enter the screen mode (in the range 0 to 7), the filename of the saved Beeb screen, and the filename with which to save the converted screen. If this last name is omitted (by pressing Return alone) then the picture will be displayed but not saved.

```
   10 REM            >BBCscrCON
   20 REM Program    Screen Converter
   30 REM Version    A1.0
   40 REM Author     Dag Haakon Myrdal
   50 REM RISC User  July/August 1989
   60 REM Program    Subject to Copyright
   70 :
   80 MODE 0:DIM B% 30:Z%=FNaddr
   90 PRINT"BBC TO ARCHIE SCREEN FILE CO
NVERTER"'''
  100 *CAT
  110 PRINT
  120 INPUT"Screen Mode (0-7) :"Mode%
  130 INPUT"Filename to convert :"file$
  140 INPUT"Filename of file to create o
r <Return> :"arcfile$
  150 MODE Mode%
  160 PROCread(file$,Mode%)
  170 IF arcfile$<>"" THEN OSCLI"SCREENS
AVE "+arcfile$
  180 END
  190 :
  200 DEF FNaddr
  210 !B%=148:B%!4=-1
  220 SYS"OS_ReadVduVariables",B%,B%+&10
  230 =B%!&10
  240 :
  250 DEF PROCread(file$,Mode%)
  260 offset%=0
  270 handle%=OPENIN file$
  280 ON ERROR CLOSE#handle%:REPORT:END
  290 IF Mode%=7 THEN
  300 VDU 30,23,16,1|
  310 FOR F%=1 TO 1000
  320 G%=BGET#handle%
  330 IF G%<32 THEN G%+=128
  340 VDU G%
  350 NEXT:VDU 30
  360 ELSE
  370 WHILE NOT EOF#handle%
  380 FOR Y%=0 TO 79
  390 FOR X%=0 TO 7
  400 addr%=Z%+(X%*80)+offset%+Y%
  410 ?addr%=FNGetInv(Mode%)
  420 NEXT:NEXT
  430 offset%=offset%+80*8
  440 ENDWHILE
  450 ENDIF
  460 CLOSE#handle%
  470 ON ERROR OFF
  480 ENDPROC
  490 :
  500 DEF FNGetInv(M%)
  510 out%=0:in%=BGET#handle%
  520 CASE M% OF
  530 WHEN 0,3,4,6
  540 IF(in% AND 1) out%=out% OR 128
  550 IF(in% AND 2) out%=out% OR 64
  560 IF(in% AND 4) out%=out% OR 32
  570 IF(in% AND 8) out%=out% OR 16
  580 IF(in% AND 16) out%=out% OR 8
  590 IF(in% AND 32) out%=out% OR 4
  600 IF(in% AND 64) out%=out% OR 2
  610 IF(in% AND 128) out%=out% OR 1
  620 =out%
  630 WHEN 1,5
  640 IF(in% AND 128) out%=out% OR 1
  650 IF(in% AND 8) out%=out% OR 2
  660 IF(in% AND 64) out%=out% OR 4
  670 IF(in% AND 4) out%=out% OR 8
  680 IF(in% AND 32) out%=out% OR 16
  690 IF(in% AND 2) out%=out% OR 32
  700 IF(in% AND 16) out%=out% OR 64
  710 IF(in% AND 1) out%=out% OR 128
  720 =out%
  730 WHEN 2
  740 IF(in% AND 128) out%=out% OR 1
  750 IF(in% AND 32) out%=out% OR 2
  760 IF(in% AND 8) out%=out% OR 4
  770 IF(in% AND 2) out%=out% OR 8
  780 IF(in% AND 64) out%=out% OR 16
  790 IF(in% AND 16) out%=out% OR 32
  800 IF(in% AND 4) out%=out% OR 64
  810 IF(in% AND 1) out%=out% OR 128
  820 =out%
```

BRAINSOFT'S MULTIPLE I/O PODULE

Reviewed by Lee Calcraft

The spec of Brainsoft's new podule for the Archimedes is truly impressive. On just one half-width podule they have fitted:

- A mono video digitiser
- A stereo sound sampler
- An analogue port compatible with that on the BBC Micro
- A pair of RS 232 ports and three ROM sockets

And all this costs just £107 +VAT.

The board is extremely well-made, and is easily fitted into the machine, assuming that you already have a podule backplane. It has three sockets at the rear, a BNC for video input, a standard 15 pin D connector for audio and analogue input, and a non-Acorn standard 25 pin D connector for the RS 232 ports. The driver software for the podule is held in a 32K ROM on the board itself, and this is automatically installed when the machine is powered up. Unfortunately, although this provides a number of low level SWI calls, only four star commands are provided.

The present version of the Multi-Podule I/O board (version 1.2) as marketed by Brainsoft, does not in fact meet with the full specification above in the following respects:

- No disc-based software is provided to run any part of the podule.
- The ROM filing system is not implemented, so that ROMs may not be used with this board.
- Sound sampling is implemented in mono only.
- Only one of the two RS 232 ports is functional.

In the near future Brainsoft are hoping to release a new version of the board which will overcome these shortcomings. Upgrades will be available for a small sum.

THE VIDEO DIGITISER

Two star commands are provided for using the digitiser:

```
*BVidSamp12   and
*BVidSamp15
```

Both require a filename, and will grab a frame (in reality an amalgamation of 100 frames - see later), display it on the screen (mode 12 or mode 15 depending on the call) and save it to disc. It is a shame that there is no star command to perform the task without saving to disc - though you can achieve this easily with a pair of SYS (or SWI) calls:

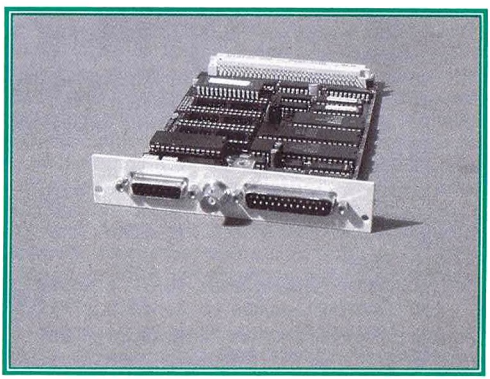
```
SYS &408C0:SYS &408C2
```

for mode 12, and:

```
SYS &408C0:SYS &408C1
```

for mode 15.

In use I found Brainsoft's podule software to be somewhat unstable. If you call for a frame to be grabbed, and no signal is present, instead of the "No video signal" message which you get from Watford's digitiser, the machine just hangs. You can't even restore it with Ctrl-Break. Ctrl-Reset will get you out, but after this the podule's star commands and SWIs cease to be recognised. Only by switching off the computer can you make things work again. I found that the software hung up on other occasions too. If you tell it to perform a mode 15 frame-grab, but the computer is in mode 12, it will also hang up, and so on.



Brainsoft's multiple I/O podule

The picture quality of the grabbed image was also most disappointing. In order to keep the cost of the unit to a minimum, Brainsoft have used a relatively slow D/A (digital to analogue) converter to scan the video image - in fact the same converter is used for the video digitiser, the sound sampler and the analogue port, so they can't be used simultaneously. There is also no RAM on the board (Watford's digitiser has 160K of on-board RAM for fast frame storage).

Because it is a relatively slow device, the A/D is not capable of scanning a single video frame in real time. It actually scans around 100 frames to build up its image. This takes some two seconds or so, and for the very reasonable price of the board, two seconds seems quite acceptable. But there is a problem inherent in this approach which may be insoluble. In all tests which I carried out, both with a VHS-C video camera, and with a good quality video recorder (used in freeze-frame mode to halt the picture for 2 seconds), all images were very noisy. This manifested itself as distortion on all horizontal lines of the picture. In spite of advice from Brainsoft, I was not able to overcome this.

THE ANALOGUE PORT

The Brainsoft analogue port simulates that on the BBC Micro, or on Acorn's Archimedes I/O podule. You can access it using Basic's ADVAL function, so that the board's dearth of star commands is no drawback here. In tests, I found that although the port caused values to be returned by ADVAL, its output was far from stable. A short program to print out the voltage at the analogue input at intervals of 0.1 seconds gave the following sequence of of readings with a stable resistive source:

```
14592, 15872, 15616, 14336
```

and at the low end, fluctuations were even greater:

```
512, 256, 2560, 2560, 512
```

and so on. The same input applied to the analogue port of a BBC Master gave constant and unchanging values.

SOUND SAMPLER

The podule's sound sampler will digitise audio signals input at the A/D connector. Two star commands are provided for this purpose:

```
*BSoundSamp
```

wrongly given in the manual as *SoundSamp, and

```
*BSoundPlay
```

confusingly given in the manual as *PlaySound.

When you get the name right, both commands take a filename as parameter, and store a 4 second sample to disc or replay it, respectively. It is a shame that there are no star commands for taking samples of other lengths or for replaying them without saving them to disc. There is also a problem with the RMA. The sample is built up in the RMA, and although the manual doesn't explain this, you need to configure some 250K for this purpose. What is really annoying is that the RMA is not cleared afterwards, so that if you repeat the command *BSoundSamp, it will fail with a *No room in RMA* message.

If you want to take samples of different lengths, you can use one of the SYS calls provided, though I failed to get these to behave properly. During sampling the only way to escape is to use Ctrl-Reset (Ctrl-Break would not work), and this again caused the computer to stop recognising any of the podule commands. It was as if the Arc had had enough of this podule, and I must admit to feeling the same way about it myself.

As to the quality of the sampled sound itself, this was seriously impaired by a high level of white noise, which was itself distorted.

CONCLUSION

I would like to think that the board reviewed here was faulty, and that there is nothing fundamentally wrong with the product itself. However, by no means all of the shortcomings of this product can be put at the door of a single board fault. The system software is obviously in need of an overhaul, and I have my doubts about the feasibility of scanning a video frame by sampling from 100 frames using a relatively slow A/D device. In such a set-up, each frame must be identical to the last, but more importantly the timing of each individual digital sample (i.e. the timing applied to the sample-and-hold device) will be absolutely crucial if image distortion is to be avoided.

However, Brainsoft's attempt to provide a useful cluster of functions on a single board and at a very reasonable end-user price is laudable. If they can supply us with a new version with these problems fixed, we will report back.

Product	**Multiple I/O Podule**
Supplier	**Brainsoft Ltd** **106 Baker Street,** **London W1M 1DF.** **Tel. 01-486 0321**
Price	**£123.05 inc. VAT + £3 p&p**

RU

EXPANSION CARDS FOR THE ACORN ARCHIMEDES COMPUTER SYSTEM

IEEE488 INTERFACE a full implementation of the standard for automatic test and measurement systems

16 BIT PARALLEL I/O two 16 bit input or output ports with handshake lines for digital control applications

DUAL RS423 SERIAL INTERFACE for communicating with two additional RS423 or RS232 devices eg printers, plotters, instruments, etc

12 BIT ANALOGUE I/O in development

All the above high performance expansion cards are supplied with high level software for ease of use and a comprehensive user guide.

Take advantage of Intelligent Interfaces' expertise and purchase a complete Archimedes Computer System.

Officially appointed Acorn Scientific Dealer.

Intelligent Interfaces Ltd
43b Wood Street
Stratford-upon-Avon
Warwickshire
CV37 6JQ
Tel: 0789 415875
Telex: 312242 MIDTLX G

A Basic Program Cross-Referencer

Keep your Basic programs under control with this magazine disc utility from Barry Christie.

Because of space constraints, the cross-referencer program is included on the magazine disc only.

The LVAR command in Basic V can be very useful in debugging a program. It will list all the variables defined during the execution of a program, along with the names of any functions and procedures called, and any libraries loaded. However, LVAR is totally dynamic - a variable is only listed if it has actually been defined during execution. Another debugging tool is LIST IF. This will list all the lines containing the specified keyword, variable, etc. For example:

```
LIST IF PRINT
```

While LVAR and LIST IF can prove invaluable, it is sometimes very useful to be able to list out in one go all the variables, procedures etc., together with the numbers of all the lines referencing them. This is exactly what the cross-referencer presented here does.

USING THE CROSS-REFERENCER

The cross-referencer takes the form of a relocatable module, and this is supplied 'ready to load' on the monthly disc. The disc menu gives full instructions for loading the module. Also contained on the disc is the source code for the module, and this is fully documented making it easy to follow.

Once loaded, the cross-referencer module offers two commands:

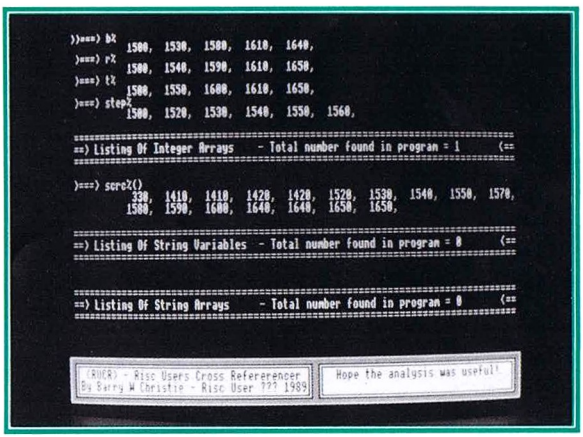
```
*XREFD    and
*XREFP
```

These two commands are basically the same, except that the first displays the output on the screen, while the seconds prints the results as well. Used with no parameters, the current Basic program is cross-referenced, while adding an optional filename will load the named program and then cross-reference it. For example:

```
*XREFP BasicProg
```

The output produced by the cross-referencer is split into eight categories:

Procedures
Functions
Real variables
Real arrays
Integer variables
Integer arrays
String variables
String arrays



Cross-referencer in action

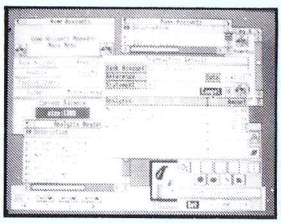
For each category, the total number of objects of that type is printed, followed by the name of each object and the numbers of the lines on which it occurs. The diagram shows an example of the display. During the process, Escape can be used to abort the operation. Pressing and holding Space will freeze the display until the key is released.

Obviously, the cross-referencer in itself cannot debug a program. However, armed with the list of occurrences, it should be relatively straightforward to pick out possible 'problem lines'.

For details of ordering the magazine disc, refer to the inside back cover. RU

ARCHIMEDES SOFTWARE

HOME ACCOUNTS
£49.95



Keep your home or club finances in order. Full control of multiple bank accounts including credit cards and building societies. Budgeting facility to assist cashflow prediction. Graphs of budgets and expenditure. Caters for standing orders, transfers, statement reconciliation, etc. (Multi-tasking if using RISC OS.)

ANCESTRY
£79.95

The ultimate in computerised genealogy! for use in families or pedigree animals. Use added card index for extra data or photos of individuals. Demo file contains illustrated Royal Family tree. Vertical or horizontal printout of family tree. For home, hobby or educational use.

(RISC OS compatible)

ATELIER
£99.95

Professional quality art package. Probably the most sophisticated sprite editor available. Animation of image sequences with Quantel-style graphic effects. Anti-alias squashing, rotation of areas into any quadrangle, and sprite and brush patterning. Further features allow TV-style techniques to even wrap a picture around a 3D surface. (Multi-tasking if using RISC OS.) (To be released soon).

Prices inclusive of VAT.

Access & Barclaycard accepted

M·I·N·E·R·V·A

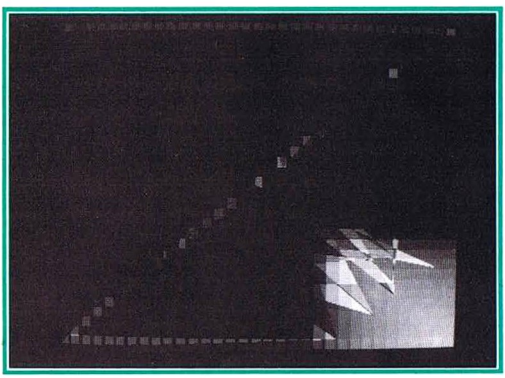
69 SIDWELL STREET, EXETER. EX4 6PH Tel: (0392) 437756 Fax: (0392) 421762

Archimedes Visuals

Graham Veitch produces some very smart effects by programming the VIDC Chip.

This program provides a set of procedures which are useful for creating special effects in any mode from 0 to 17 by reprogramming the Video Controller chip (VIDC). Although the VIDC is a complex device, the procedures listed here should make it easy to create two particular types of effect.

Firstly, you can move the outer edge of the border around the screen to any specified rectangle. If the border is moved to within the display area, it forms a movable window on the display. This is the first part of the demonstration supplied. Alternatively, you can "wrap up" the display area so that it appears in any specified rectangle on the screen, with border round about (this is the second part of the demonstration).



The program in operation

To use the facilities provided by this program in your own programs, you must include PROCassemble, PROCscrinit and PROCscr. You will probably also wish to include PROCscrto.

Before using PROCscr or PROCscrto, you must first call PROCassemble to assemble the machine code routine which writes to the VIDC, then PROCscrinit which reads information about the current mode and sets up parameters in the machine code. This procedure should therefore be called with the screen mode which you intend to use already selected. It also sets up the global variables scrc% and pixshift% which are used by PROCscr and PROCscrto. Therefore you should not use these names for variables you define in the rest of the program.

PROCscr forms the basis of all the other procedures and moves either the border or display to within a given rectangle. It can also put the new co-ordinates into the array scrc%, which is primarily for the use of PROCscrto. The parameters for PROCscr are:

```
PROCscr(left,bottom,right,top,select,set)
```

The first four give the co-ordinates in normal graphics units of the new rectangle. *Left* and *top* refer to a pixel just inside the top-left corner of the rectangle, *right* and *bottom* to a pixel just outside the bottom-right corner. Co-ordinates are measured, as usual, from the bottom-left corner of the normal display area.

In most modes, the horizontal co-ordinates will be numbered from 0 to 1279 for the default area, but in modes 16 and 17, as there are 1056 pixels horizontally, the numbering is from 0 to 1055 for the default area. With negative co-ordinates, or co-ordinates greater than the usual maximum, the 'screen' can be expanded.

One additional point is that the number of bits in a line of the display area (i.e. the number of pixels multiplied by the bits per pixel) must be a multiple of 16 to achieve a steady display.

The fifth parameter, *select*, should be zero if you wish to move the outer edge of the border, and one if you wish to move the display area. The final parameter, *set*, should be TRUE if you wish to store the new co-ordinates in scrc%, FALSE if you do not. This should normally be TRUE, so that PROCscrto can correctly calculate the steps it requires to take. PROCscrto itself sets this parameter to FALSE as it performs its own updating.

PROCscrto will smoothly move the current border or display rectangle to a new one, in a specified number of steps. For this it requires correct values in the array scrc%. It would therefore be usual to make a call to PROCscr,

with set=TRUE before using PROCscrto. PROCscrto takes very similar parameters to PROCscr. They are:

PROCscrto(left,bottom,right,top,select,steps)

The first five parameters have the same function as in PROCscr. The last specifies the number of steps required to move from the old to the new rectangle. The steps are made once every frame (i.e. fifty per second) by use of the WAIT command.

Three procedures are included to demonstrate the use of PROCscr and PROCscrto. These are PROCswap, which contracts the screen to any point, then swaps to a different bank before expanding the screen, and PROCbounce and PROCstretch which provide two possible routines for swapping between screens.

The best way to understand how to create your own effects is to modify these procedures. If you should lose the screen display, escape from the program, then type a mode change command; if this fails, which is unlikely, pressing Reset will always work. Alternatively, the final procedure in the demonstration shows one method of resetting the screen to a reasonable condition.

With these procedures, a large number of effects are available: the only limit is your own imagination in their application.

```
 10 REM            >VidcProg11
 20 REM Program    VIDC Programmer
 30 REM Version    A 0.11
 40 REM Author     Graham Veitch
 50 REM RISC User  July/Aug 1989
 60 REM Program    Subject to Copyright
 70 :
 80 MODE 12:OFF
 90 PROCassemble
100 PROCscrinit
110 ON ERROR PROCerror:END
120 PROCscreen
130 REPEAT
140 *Fx 112 1
150 *Fx 113 1
160 FOR bord%=0 TO 1
170 PROCscr(-96,-64,1376,1088,0,TRUE)
180 PROCscr(0,0,1280,1024,1,TRUE)
190 FOR vpos%=0 TO 1
200 FOR hpos%=0 TO 1
210 PROCswitch1(hpos%*1280,vpos%*1023,
(vpos%+hpos%) MOD 2+1)
220 REPEAT UNTIL INKEY(150)
230 NEXT hpos%
240 IF vpos%=0 THEN PROCbounce(1) ELSE
PROCstretch(2)
250 REPEAT UNTIL INKEY(150)
260 NEXT vpos%
270 NEXT bord%
280 UNTIL FALSE
290 END
300 :
310 DEF PROCassemble
320 LOCAL code%,opt%
330 DIM code% &7F,scrc%(3,1)
340 FOR opt%=0 TO 3 STEP 3
350 P%=code%
360 [
370 OPT opt%
380 .screensizer
390 SWI 22
400 MOVNV R0,R0
410 MOV R7,#&03400000
420 TEQ R4,#0
430 LDREQ R5,hbstart
440 LDRNE R5,hsstart
450 LDR R6,vstart
460 ADD R0,R5,R0,ASR #1
470 ADD R2,R5,R2,ASR #1
480 ADD R1,R6,R1
490 ADD R3,R6,R3
500 ADD R0,R0,#&88000000
510 ADD R1,R1,#&B4000000
520 ADD R2,R2,#&94000000
530 ADD R3,R3,#&A8000000
540 ADD R0,R0,R4:SUB R1,R1,R4
550 SUB R2,R2,R4:ADD R3,R3,R4
560 STMIA R7,{R0-R3}
570 TEQP PC,#0:MOVNV R0,R0
580 MOVS PC,R14
590 .hbstart EQUD 0
600 .hsstart EQUD 0
610 .vstart EQUD &98000
620 ]
630 NEXT
640 ENDPROC
650 :
660 DEF PROCscrinit
670 LOCAL bytes%,pix%,pixbits%,hstart%
,vstart%
680 SYS 53,MODE,6 TO ,,bytes%
690 SYS 53,MODE,9 TO ,,pixbits%
700 pix%=bytes%*8>>>pixbits%
710 CASE pix% OF
720 WHEN 1056:hstart%=&A6:pixshift%=0
```

```
  730 WHEN 640:hstart%=&84:pixshift%=1
  740 WHEN 320:hstart%=&42:pixshift%=2
  750 WHEN 160:hstart%=&21:pixshift%=3
  760 ENDCASE
  770 !hbstart=hstart%<<14
  780 CASE pixbits% OF
  790 WHEN 0:!hsstart=hstart%-9<<14
  800 WHEN 1:!hsstart=hstart%-5<<14
  810 WHEN 2:!hsstart=hstart%-3<<14
  820 WHEN 3:!hsstart=hstart%-2<<14
  830 ENDCASE
  840 ENDPROC
  850 :
  860 DEF PROCscreen
  870 LOCAL loop%,char%,x%,y%
  880 VDU 19,0,24,&F0,&80,&80
  890 *Fx 112 1
  900 *Fx 113 1
  910 COLOUR 0,4
  920 CLS
  930 *Help
  940 *Help FileCommands
  950 *Fx 112 2
  960 *Fx 113 2
  970 CLS
  980 MOVE RND(1280)-1,RND(1024)-1
  990 FOR loop%=0 TO 31
 1000 GCOL 3,RND(7)
 1010 PLOT 85,RND(1280)-1,RND(1024)-1
 1020 NEXT
 1030 ENDPROC
 1040 :
 1050 DEF PROCbounce(scr%)
 1060 PROCscr(0,0,1280,1024,bord%,TRUE)
 1070 PROCscrto(0,992,32,1024,bord%,32)
 1080 PROCscrto(1248,992,1280,1024,bord%
,32)
 1090 PROCscrto(0,0,32,32,bord%,32)
 1100 PROCscrto(1248,0,1280,32,bord%,32)
 1110 SYS 6,112,scr%
 1120 SYS 6,113,scr%
 1130 PROCscrto(0,0,1280,1024,bord%,32)
 1140 ENDPROC
 1150 :
 1160 DEF PROCswitch1(x%,y%,scr%)
 1170 PROCscr(0,0,1280,1024,bord%,TRUE)
 1180 PROCscrto(x%,y%,x%+4,y%+4,bord%,32
)
 1190 SYS 6,112,scr%
 1200 SYS 6,113,scr%
 1210 PROCscrto(0,0,1280,1024,bord%,32)
 1220 ENDPROC
 1230 :
 1240 DEF PROCstretch(scr%)
 1250 PROCscr(0,0,1280,1024,bord%,TRUE)
 1260 PROCscrto(0,1008,16,1024,bord%,32)
 1270 PROCscrto(0,1008,320,1024,bord%,8)
 1280 PROCscrto(960,1008,1280,1024,bord%
,24)
 1290 PROCscrto(1264,1008,1280,1024,bord
%,8)
 1300 PROCscrto(1264,768,1280,1024,bord%
,8)
 1310 PROCscrto(1264,0,1280,256,bord%,24
)
 1320 PROCscrto(1264,0,1280,16,bord%,8)
 1330 SYS 6,112,scr%
 1340 SYS 6,113,scr%
 1350 PROCscrto(0,0,1280,1024,bord%,32)
 1360 ENDPROC
 1370 :
 1380 DEF PROCscr(A%,B%,C%,D%,E%,set%)
 1390 IF B%>=D% THEN D%=B%+4
 1400 IF set% THEN
 1410 scrc%(0,E%)=A%:scrc%(1,E%)=B%
 1420 scrc%(2,E%)=C%:scrc%(3,E%)=D%
 1430 ENDIF
 1440 A%=A%<<14-pixshift%:B%=(1023-B%) D
IV 4<<14
 1450 C%=C%<<14-pixshift%:D%=(1023-D%) D
IV 4<<14
 1460 E%=E%<<26
 1470 CALL screensizer
 1480 ENDPROC
 1490 :
 1500 DEF PROCscrto(l%,b%,r%,t%,E%,step%
)
 1510 LOCAL ls,rs,ts,bs,loop%
 1520 ls=(l%-scrc%(0,E%))/step%
 1530 bs=(b%-scrc%(1,E%))/step%
 1540 rs=(r%-scrc%(2,E%))/step%
 1550 ts=(t%-scrc%(3,E%))/step%
 1560 FOR loop%=1 TO step%
 1570 l%=scrc%(0,E%)+loop%*ls
 1580 b%=scrc%(1,E%)+loop%*bs
 1590 r%=scrc%(2,E%)+loop%*rs
 1600 t%=scrc%(3,E%)+loop%*ts
 1610 PROCscr(l%,b%,r%,t%,E%,FALSE)
 1620 WAIT
 1630 NEXT
 1640 scrc%(0,E%)=l%:scrc%(1,E%)=b%
 1650 scrc%(2,E%)=r%:scrc%(3,E%)=t%
 1660 ENDPROC
 1670 :
 1680 DEF PROCerror
 1690 PROCscr(-96,-64,1376,1088,0,TRUE)
 1700 PROCscr(0,0,1280,1024,1,TRUE)
 1710 ON:REPORT
 1720 PRINT " at line ";ERL
 1730 ENDPROC
```

RU

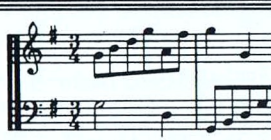
ARCHIMEDES SOFTWARE

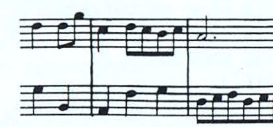
Disc 1 EMACS. Good editor with language. Now includes full C source code.
Disc 2 MicroSpell. 43,000 word spelling checker for EMACS or standalone use.
Disc 4 XLISP. Object oriented version of LISP with C source.
Disc 5 C Toolkit. Over 20 programs for C programmers. grep, make, awk ...
Disc 6 Kermit. Archimedes, BBC and MSDOS versions of file transfer prog.
Disc 7 WIMP Chess. Good standard Wimp chess game with lots of features.
Disc 8 Cross Star. Wimp based crossword solver with huge dictionary.
Disc 10 More C Tools. yacc and lex with examples and C source. diff, ctags...
Disc 11 Little SmallTalk. Original object oriented language with C source.
Disc 14 Gnu Tools #1. High quality tools with C source. egrep, sed, awk and diff.
Disc 15 Clip Art #1. Over 50 pictures to paste into DTP/ArcDraw etc. files.
Disc 16 Spark. Super RISC OS WIMP based multitasking file archiver. Also archives directories. Compatible with PC arc and PKARC files.
Disc 17 U Tools. Tar, Compress, Less and Sort with C source code. UNIX Tar archiver extended for the Archimedes with compress file squasher.

Each disc is £5.99 inc. Buy four claim one free!

David Pilling, P.O. Box 22, Thornton Cleveleys, Blackpool. FY5 1LR.

Free Air Mail delivery on overseas orders. ***More details available on request.***



EUCORN!

- EUCORN is a revolutionary new generation of musical instrument.
- With it one can perform prescribed pieces of music like a musician.
- Real time control of musical interpretation during performance.
- Runs on Archimedes 310, with amplifier & speakers: printer desirable.
- Disk contains performance program (both BASIC and ABC compiled versions) a music input program, a repertoire of classical pieces (over 2 hours playing time) and detailed 70+ page User Guide as a print file.
- Customers welcome to contact author at any time.
- There is **NO CHARGE** for EUCORN: after receipt & testing, it is hoped that customers will donate a suitable sum directly to any recognised charity.
- To obtain the package, please send a 1st class UK postage stamp, a self-addressed (UK addresses **ONLY**) envelope & a 800k formatted 3.5" disk. Software will be copied onto your disk & posted back to you the same day.

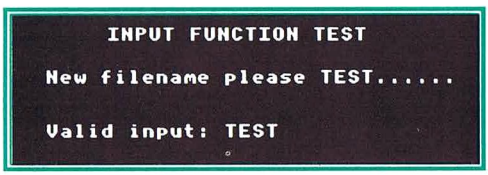
STELLAR ENTERPRISE
(Ronald Alpiar),
84 Dudsbury Road, Ferndown, Wimborne, Dorset BH22 8RG.
Telephone Bournemouth (0202) 575234.

ARC PROCEDURE LIBRARY

by Lee Calcraft

This month's Library features a general purpose input function.

The standard INPUT function provided by Basic is easy to use, but has a number of limitations. For example, the limit on user input is 238 characters, so if the user holds down a key, even the best laid out screen will be spoiled. There is also no restriction on which keyboard characters are accepted, the mouse is completely ignored, and so on.



The function supplied here solves these problems, and a few more besides. It takes six parameters, as you can see from the definition. The first is the maximum length of string that the user may supply. The second and third define the range of ASCII characters that the routine will accept. The fourth is a string containing any characters within the above range which should be ignored (use "" if there are none). The fifth parameter is the so-called "paint" character which will mark the position of each character of the input string before it is entered (e.g. "-" will cause a row of dashes to be used, or " " will give a blank space). Finally the sixth parameter indicates whether the input should be echoed to the screen (TRUE) or not (FALSE). You might set this to FALSE on a password input to avoid the password appearing on the screen.

The function returns the string input by the user. The user must terminate his input either by pressing the Return key, or the Select button. If on the other hand he uses the Menu or Adjust buttons at any time, this will immediately cancel his input, and to signify this the string returned by the function will be the single character CHR$127 (ASCII delete).

If the user terminates the string using the mouse, the variable *exitchoice%* will be set to the mouse button number. This enables you to distinguish between cancelling with the Menu and the Adjust button, which may be important for some applications.

If you run the program you can see the input function in use. In the example the length of input is restricted to 10 characters (as it might be if used for inputting filenames), and certain characters, including brackets, colons and so on, are excluded.

You will see that the value returned by the function (line 130) is checked for validity before the result is used. Also the variable exitchoice% is checked if the user cancelled his input. If he used Adjust then a further input is sought, while if he cancelled with the Menu button, the program terminates.

Finally, although the function is designed primarily for the input of text strings, it is relatively easy to customise. To make it yield a numeric result, use:

```
result=VAL(FNinput(len%,45,69,"/:;<=>?@ABC
D","-",TRUE))
```

For 4-digit hex input, use:

```
result=EVAL("&"+FNinput(4,48,70,":;<=>?@",
"-",TRUE))
```

```
   10 REM            >Prc4-1-3
   20 REM Procedure/Function Library
   30 REM Version    A 0.3
   40 REM Author     Lee Calcraft
   50 REM RISC User July/Aug 1989
   60 REM Copyright Lee Calcraft
   70 :
   80 MODE12
   90 PRINTTAB(10,7)"INPUT FUNCTION TEST
"
  100 REPEAT
  110 PRINTTAB(0,10)"New filename please
 ";
  120 result$=FNinput(10,48,122,":;<=>?@
[\]^£",".",TRUE)
  130 IF result$<>CHR$127 THEN
  140 PRINT'"Valid input: ";result$;SPC(
10)
  150 ELSE VDU7:PRINT'"Cancelled entry";
SPC(10)
  160 ENDIF
  170 UNTIL exitchoice%=2
  180 END
  190 :==================================
  200 REM Input function - Returns CHR$
127 if input invalid
  210 REM exitchoice% holds mouse button
 at exit
  220 :
```

```
  230 DEFFNinput(len%,lochr%,hichr%,excl
ude$,paint$,echo%)
  240 LOCAL n%,paint%,inputok%,quitinput
%
  250 quitinput%=FALSE:inputok%=FALSE
  260 input$="":n%=0:exitchoice%=0
  270 paint%=ASC(paint$)
  280 PRINT STRING$(len%,paint$);STRING$
(len%,CHR$8);
  290 REPEAT
  300 PROCmousewait(-24)
  310 IF z%>0 THEN
  320 IF z%=4 THEN inputok%=TRUE ELSE qu
itinput%=TRUE
  330 ELSE
  340 CASE TRUE OF
  350 WHEN w%=13:inputok%=TRUE
  360 WHEN w%=127 AND n%>0:n%-=1
  370 input$=LEFT$(input$,n%):VDU8,paint
%,8
  380 WHEN w%>=lochr% AND w%<=hichr% AND
 INSTR(exclude$,CHR$w%)=0
  390 IF n%>=len% THEN
  400 VDU7
  410 ELSE input$+=CHR$(w%):n%+=1
  420 IF echo% THEN VDUw% ELSE VDU9
  430 ENDIF
  440 ENDCASE
  450 ENDIF
  460 UNTIL quitinput% OR inputok%
  470 IF quitinput% THEN input$=CHR$127
  480 exitchoice%=z%:PRINT:
  490 PROCmousewait(0)
  500 =input$
  510 :================================
  520 DEFPROCmousewait(n)
  530 REM 0-7: Wait for this mouse state
  540 REM neg: Wait for zero first
  550 REM 8:   Wait for any button
  560 REM 16:  Wait for keypress
  570 LOCAL anypress,inkey,m
  580 *FX21,9
  590 IF n<0 THEN PROCmousewait(0):n=ABS
(n)
  600 IF (n AND 16)>0 THEN inkey=TRUE:*F
X21
  610 IF (n AND 8)>0 THEN anypress=TRUE
  620 m=n AND 7:w%=-1
  630 REPEAT
  640   MOUSE x%,y%,z%
  650   IF inkey THEN w%=INKEY(0)
  660 UNTIL (z%=m AND NOT (n>7 AND m=0))
 OR (anypress AND (z%>0)) OR (inkey AND
(w%>-1))
  670 ENDPROC
```

RU

ARCHEFFECT

FX
VISUAL
EFFECTS

ARCHEFFECT is an easy to use image manipulation package. By the use of simple *Commands you will be able to quickly produce spectacular visual effects, similar to those seen on TV. It also allows for the easy use of fonts.

The module has been specifically designed for use in the high resolution graphics modes to take full advantage of the potential of the Archimedes.

In addition to the many image manipulation commands the module provides an alternative method of using the powerful font manager provided on the Archimedes, bypassing the complex 'SYS' commands previously necessary.

ARCHEFFECT - £24.95 inc VAT and p&p

Please send me ..... copies of Archeffect at £24.95 each.
I enclose a cheque or postal order for the sum of £.....

Please make cheques or postal orders payable to FX.
FX, 207 South Avenue, Southend-on-Sea, Essex SS2 4HT.
(For further details send SAE)

BASIC SQUEEZE

When the Archimedes was first launched, with half a mega-byte of RAM or more, it looked as if there would never be any need to squeeze Basic programs to fit into the machine. But the Arc is also a voracious eater of RAM, and with screen, sprite and font space allocated, let alone RAM discs, you can soon run out of memory. And if you are trying to multi-task with less than 2 Mbytes of RAM, things feel distinctly tight. So a facility to squeeze Basic programs could be an attractive proposition.

The program listed here will do just that. It has two options: it will remove REMs and/or spaces from any program. The program is itself written in Basic to make it easier to follow, and to customise. This makes it a little slow on very large programs, but it is not likely that you will use it very frequently.

To use the program, type it in carefully, and save it to disc. Next create a short program as follows:

```
10*KEY1 PAGE=TOP+&200|M CHAIN"Squeeze"|M
20*KEY2 PAGE=&8F00|MOLD|M
```

and save this under the name "K" for keys. Type *K to set up the function keys, then load the program that you wish to compact (e.g. type LOAD"program").

If you now press function key F1, the Squeeze program will be loaded into memory above the target program, and run. It will prompt you to select REMs and/or spaces as targets for removal, and the program will proceed, indicating its progress as it goes. When it has finished, you will be able to list, run or save the compressed program in the normal way.

Function key F2 is only included for debugging purposes. If the Squeeze program stops with an error before finishing (as it would if the Squeeze program were mis-typed), pressing function key F2 will allow you to list the target program; though in this case it is quite likely that you would get a *Bad program* error with the target program. Incidentally, you should avoid pressing Escape while the compressor is at work for the same reason - the target program will be corrupted.

REMS

If you have opted not to remove REM lines, the program will leave REM lines exactly as it finds them, since removing spaces from such lines would not necessarily be helpful. When REM lines *are* removed, only those lines which begin with a REM (optionally preceded by spaces and/or colons) will be affected. When the REM option is set, the program generates a table of line number references (e.g. GOTO 100 etc.), and REMs which are pointed to by such lines are not removed (though a message is given in such cases).

SPACES

When removing spaces, the program will of course avoid removing spaces from anything between quotation marks. It will also leave in the vital space between statements such as:

```
IFflag b%=TRUE
```

and will compress DATA statements without the loss of significant spaces. Because the program removes all redundant spaces from the target program, you need to be very careful how you treat a compressed program. You will not be able to edit it in the normal way once spaces have been removed - either by using cursor editing, or an editor such as Twin or the Basic Editor, and for that reason it is advisable always to keep an uncompressed copy of your work.

HOW IT WORKS

Essentially the program works by scanning through a Basic program in memory, and shortening it where appropriate. To do this it must take careful account of the way in which Basic programs are stored.

By way of illustration, consider the following short program:

```
10 REM A
20 REM
```

This would be stored in RAM as the following hex sequence:

```
0D
00 0A 08 20 F4 20 41 0D
00 14 06 20 F4 0D
FF
```

The program starts with byte &0D (13 decimal) - a carriage return. This is followed by the number of the first line, given as a 16 bit

number (high byte first), then a byte giving the length of the line (&08 in the first line). There then follows a sequence giving the contents of the line itself, with most keywords being represented as one or two token bytes. In this case &20 (a space) followed by &F4 (the token for REM) plus &20 (a space) and &41 (ASCII A). Finally, a carriage return (&0D) signifies the end of the first line. This coding continues until the end of the program is reached, signified by &0D &FF.

In the present program, two complete scans of the target are made (unless "spaces only" is selected), the first scan is carried out by PROClinerefs. Here the object is to build up a table of line references (GOTO etc.) which will later be checked before REM lines are removed.

The main program scan is carried out by PROCscan. This picks up the length of each program line, storing it in the variable *len%*. The line is then checked to see if it is a REM line (does FNgetword=&F4?). If REMs are to be removed, PROCskip moves the pointers so as to delete the REM and close up the space; otherwise PROCcopy is called which copies the line across.

Spaces operate in a similar way, except that for each space found, FNremove is used to check whether the characters either side of the space suggest that it is removable or not. The process continues until the end of the program is reached, when the final bytes &0D &FF are copied over.

```
   10 REM            >Squeeze
   20 REM Program    Program Compressor
   30 REM Version    A 0.2
   40 REM Author     Lee Calcraft
   50 REM RISC User  July/August 1989
   60 REM Program    Subject to Copyright
   70 :
   80 page%=&8F00
   90 MODE12
  100 PRINT"P R O G R A M   C O M P R E
S S O R"
  110 PRINT''"Remove REMs ?    ";
  120 dorems%=GET=ASC"Y"
  130 IF dorems% THEN PRINT"Y" ELSE PRIN
T"N"
  140 PRINT"Remove Spaces ?  ";
  150 dospaces%=GET=ASC"Y"
  160 IF dospaces% THEN PRINT"Y" ELSE PR
INT"N"
  170 IF ?page%<>13 OR page%?1=&FF THEN
ERROR 255,"No program in memory"
  180 IF dorems% THEN PROClinerefs
  190 PROCscan
  200 startsize%=source%+len%+2-page%
  210 finalsize%=dest%+dlen%+2-page%
  220 PRINT TAB(0,12)"Initial size =",st
artsize%;" bytes"
  230 PRINT"Final size   =",finalsize%;"
 bytes"
  240 PRINT"Bytes saved  =",startsize%-f
inalsize%
  250 PRINT"Lines removed=",lost%
  260 PAGE=page%
  270 *FX138,0,79
  280 *FX138,0,76
  290 *FX138,0,68
  300 *FX138,0,13
  310 END
  320 :
  330 DEFPROCscan
  340 PRINTTAB(0,8)"Line number"
  350 lines%=0:source%=page%:dest%=page%
  360 lost%=0:len%=0:dlen%=0:B%=0
  370 REPEAT
  380 source%+=len%:dest%+=dlen%
  390 len%=source%?3
  400 IF source%?len%<>13 THEN ERROR 255
,"Bad Program"
  410 lineno%=&100*(source%?1)+(source%?
2)
  420 PRINTTAB(15,8)lineno%
  430 key%=FNgetword
  440 data%=key%=&DC
  450 IF key%=&F4 THEN
  460 IF dorems% THEN
  470 IF FNnoref THEN PROCskip ELSE PROC
copy
  480 ELSE PROCcopy
  490 ENDIF
  500 ELSE
  510 IF dospaces% THEN PROCspaces ELSE
PROCcopy
  520 ENDIF
  530 lines%+=1
  540 UNTIL source%?(len%+1)=&FF
  550 dest%?dlen%=&0D
  560 dest%?(dlen%+1)=&FF
  570 ENDPROC
```

```
  580 :
  590 DEFPROClinerefs
  600 DIM gotos%(1000)
  610 PRINTTAB(0,6)"Checking line refere
nces"
  620 g%=0:source%=page%:len%=0
  630 REPEAT
  640 source%+=len%:len%=source%?3
  650 IF source%?len%<>13 THEN ERROR 255
,"Bad Program"
  660 A%=3:key%=0:inquotes%=FALSE
  670 REPEAT
  680 A%+=1
  690 key%=source%?A%
  700 IF key%=&22 THEN inquotes%=NOT inq
uotes%
  710 IF key%=&8D THEN
  720 ref%=FNgetref
  730 gotos%(g%)=ref%:g%+=1
  740 ENDIF
  750 UNTIL A%>len%-5
  760 UNTIL source%?(len%+1)=&FF
  770 ENDPROC
  780 :
  790 DEFFNgetref
  800 A%+=1:no1%=(source%?A%)<<2
  810 A%+=1:no2%=source%?A%
  820 A%+=1:no3%=source%?A%
  830 =(no2% EOR (no1% AND &C0)) OR ((no
3% EOR ((no1%<<2) AND &C0))<<8)
  840 :
  850 DEFFNnoref
  860 D%=0:match%=FALSE
  870 WHILE D%<g% AND match%=FALSE
  880 match%=lineno%=gotos%(D%):D%+=1
  890 ENDWHILE
  900 IF match% THEN PRINT TAB(0,10)"Ref
erence to line ";lineno%;" so not delete
d";SPC(6):VDU7
  910 =NOT match%
  920 :
  930 DEFFNgetword
  940 A%=3:key%=0
  950 REPEAT
  960 A%+=1:key%=source%?A%
  970 UNTIL key%<>32 OR key%>&3A OR A%>l
en%-2
  980 =key%
  990 :
 1000 DEFPROCcopy
 1010 FOR A%=0 TO len%-1
 1020 byte%=source%?A%:dest%?A%=byte%
 1030 NEXT
 1040 dlen%=len%
 1050 ENDPROC
 1060 :
 1070 DEFPROCskip
 1080 lost%+=1:dlen%=0
 1090 ENDPROC
 1100 :
 1110 DEFPROCspaces
 1120 B%=0:byte%=0:inquotes%=FALSE
 1130 last%=0:dlen%=0
 1140 FOR A%=0 TO len%-1
 1150 byte%=source%?A%
 1160 IF A%>3 AND byte%=&22 THEN inquote
s%=NOT inquotes%
 1170 skip%=A%>3 AND NOT inquotes% AND b
yte%=32 AND FNremove
 1180 IF NOT skip% THEN dest%?B%=byte%:B
%+=1
 1190 IF A%>3 AND skip%=FALSE THEN last%
=byte%
 1200 NEXT
 1210 dlen%=B%:dest%?3=dlen%
 1220 ENDPROC
 1230 :
 1240 DEFFNremove
 1250 next%=source%?(A%+1)
 1260 IF data% THEN
 1270 result%=(last%=0 OR last%=&DC OR l
ast%=&2C OR last%=&22)
 1280 ELSE
 1290 result%=FNleftok(last%) OR FNright
ok(next%)
 1300 ENDIF
 1310 =result%
 1320 :
 1330 DEFFNleftok(v%)
 1340 =v%>&7E OR v%<&21 OR v%=&28 OR (v%
>&29 AND v%<&30 AND v%<>&2E) OR (v%>&39
AND v%<&3F)
 1350 :
 1360 DEFFNrightok(v%)
 1370 IF v%=43 THEN=FNnotsave
 1380 =v%>&7E OR v%<&21 OR (v%>&29 AND v
%<&30 AND v%<>&2E) OR (v%>&39 AND v%<&3F
) OR v%=&29
 1390 :
 1400 DEFFNnotsave
 1410 E%=3
 1420 REPEAT
 1430 E%+=1:test%=source%!E%=&5641532A
 1440 UNTIL test%=TRUE OR E%>A%-3
 1450 =NOT test%
```

MACH TECHNOLOGY

MEMORY UPGRADES

Break the Megabyte barrier with Mach Technology's 2MB and 4MB expansion boards for 300 series machines.

2MByte — £340
4MByte — £520

HARD DISC SYSTEMS

Bring affordable Mass Storage to your Archimedes with our hard disc systems.

Highlights of the range include

20MBytes — £249
30MBytes — £339
40MBytes — £359

All models compatible with A305-A310 and require backplane.

HYPERBASE

An advanced database architecture based on 'cards' and 'stacks' which can contain textual and graphical information in any combination.

Cards may be connected via user defined links to allow arbitary information networks to be constructed.

Other features include powerful search and query functions across text fields, sorts and calculations, report generation and printing and the exchange of text and graphic images with other programs.

£49.95 (incl VAT)

Prices exclude VAT and carriage except where stated

MACH TECHNOLOGY LTD

75 HYDEWOOD ROAD, LITTLE YELDHAM, ESSEX CO9 4QX
Tel: (0787) 238125. Fax: (0787) 238114

DiscTree

Reviewed by Mike Williams

DiscTree is the latest software product from Mitre Software, producers of the database package Flying Start (reviewed in Volume 1 Issue 5), and DeskTop Enhancer (DTE) which runs under Arthur 1.2.

DiscTree is specifically written to run under RISC OS. For your money you get an unprotected disc and a 30 page manual. The software provides three distinct functions, a so-called *Tree Viewer* for exploring directory structures, a comprehensive back-up facility, and a related restore operation. The software generates the now standard windows and menu displays which we are used to with the Archimedes, and particularly RISC OS.

Installing DiscTree as an application on the menu bar is straightforward. Pressing the *Menu* button while pointing to DiscTree's icon brings up a menu from which a floppy or hard disc drive may be selected. A window then opens showing the complete hierarchical directory structure for that disc. This is quite informative, and DiscTree then allows you to open directly any folder in that structure.

DiscTree also provides a comprehensive 'find file' option, searching selected directories for specified file types, using wild-carded file names if you wish. This all works quite effectively, but there is one major pain to be endured. Almost every DiscTree function you choose demands that the original disc be re-inserted on two separate occasions, despite the fact that DiscTree has been installed on the menu bar. It can be effectively avoided on hard disc systems, for which DiscTree is primarily intended, by installing the software on the hard disc itself. Any subsequent access to DiscTree goes largely unnoticed.

On the plus side, DiscTree is fully multi-tasking, allowing several discs to be examined together, and file search (or other) functions to take place concurrently with any other multi-tasking application.

The back-up facility uses the same directory, file name and file type specifications as the find operation. This allows all or part of a disc to be backed up. Furthermore, files for back-up can be selected by date, so you could back up all files last changed after a particular date, for example.

I tried backing up one of my frequently used floppies. This worked just as described in the manual, but involved an inordinate degree of disc swapping, and has no real advantage over the file copy facilities available in RISC OS. However, when backing up a hard disc, DiscTree comes into its own, and will automatically create a whole set of back-up floppies as required. In addition, the complete back-up specification can be stored as a so-called script file for future reference to further automate the process.

Restoring backed-up files is also quite comprehensively handled. Files may be selected in a variety of ways (singly, by directory etc.), and unlike some systems I have seen, is flexible and does work exceedingly well. The one real chore, apart from swapping the floppies in and out, is again the recurrent necessity to re-insert the original DiscTree disc, unless installed on a hard disc.

CONCLUSIONS

If you have a wealth of valuable information stored on a hard disc system, then DiscTree is likely to prove very useful, both for regular back-ups and for its 'find file' facility, but even so the product seems rather expensive for what it has to offer. There is little value for the floppy only user, where the frequent need to re-insert the original disc is nothing but a pain.

Product	**DiscTree**
Supplier	**Mitre Software Ltd, International House, 26 Creechurch lane, London EC3A 5BA. Tel. 01-283 4646**
Price	**£49.95 inc. VAT and p&p.**

RU

ROBICO ADVENTURES
FOR THE ARCHIMEDES

RISE IN CRIME *(Archimedes 305/310/400 £29.95)*

In the vast, galaxy-wide federation, you are one of the abnormal few: unconditioned, discontented and insubordinate! Can you rid yourself of the shackles of this conformist, mind-imprisoned society? Rise in Crime boasts over 396K of text, 400 individually described locations, 150 objects and several *high resolution colour graphic screens!* Mouse driven input and an advanced parser make this a superb addition to your Archimedes collection!

FUGITIVE'S QUEST *(Archimedes 310/400 £29.95)*

As Paul Preston, respectable schoolmaster, your life is shattered when you are convicted of murder – a crime you did not commit! Set in 1952, against the stunning backdrop of southwest England, the hangman's noose awaits unless you escape and find the evidence to clear your name! Fugitive's Quest is a large adventure with about 230 locations, 80 objects and reams of well written text. An advanced parser, mouse driven input and several *full colour, photo-quality illustrations* complete the unmistakable feel of Robico quality.

ENTHAR SEVEN *(Archimedes 305/310/400 £29.95)*

It is the distant future . . . Whilst on board a tiny Interplanetary Space Hopper its orbit begins to decay! Once on the planet surface can you escape and return, again, to the freedom of the stars? Enthar Seven is a disk based mega-adventure! It has 450 locations, 80 objects, a massive vocabulary and 1200 messages! And that's just the 8-bit versions! The Archimedes version has over 140K of text and some of the *best graphics you'll ever see!*

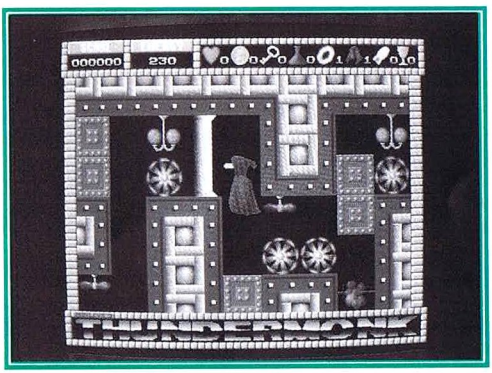
You can order by telephone using your ACCESS card, or by cutting out the coupon below. All orders are despatched first class by return.

- - - -

Peter Rochford looks at what's new on the Archimedes games front.

The Archimedes games scene continues to stay healthy with a steady trickle of new releases. No signs yet however of the much talked about 3D Pool from Firebird. Be assured, though, I'll have a review copy as soon as it becomes available.

On to the reviews. Three different types of games this month which will appeal to most tastes. First off we have the latest from Minerva, called *Thundermonk* (£14.95 inc. VAT). This is a puzzle solving arcade game where you must collect certain items in the right sequence before moving on to the next level, avoiding various hazards along the way.



Thundermonk

The game is set in a monastery and you guide a cloaked monk sporting an extraordinarily long nose, around the corridors and rooms collecting the items to gain extra energy, whilst searching for the keys to open the doors. There are lots of strange looking objects that sap your energy, but these can be destroyed by lobbing bombs at them. Very subtle!

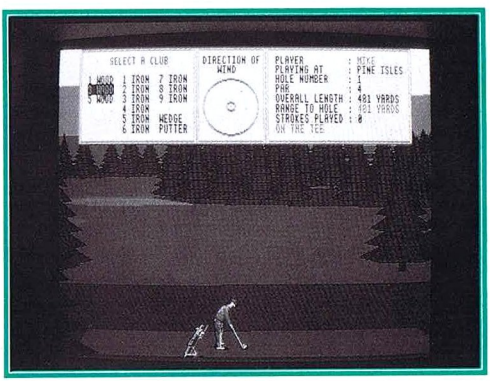
The graphics are excellent with plenty of interesting features, and the game has a smooth four-way scrolling screen. Digitised sound effects are included but mercifully can be switched off if desired.

Thundermonk is certainly an enjoyable game that quickly becomes quite addictive. The level of difficulty is pitched sensibly low at the outset, and that should make it attractive to many. Definitely recommended.

Next off we have a golf game called *Holed Out* (£19.95 inc. VAT) from a new software house called 4th-Dimension. A version of this has already been released for the Beeb and Master. The game claims to have been 'designed by enthusiasts for maximum realism'. This has been said before about other simulation-type games but as far as *Holed Out* is concerned this seems to be true. The game is mouse-driven and features several playing options, including number of players and degree of difficulty.

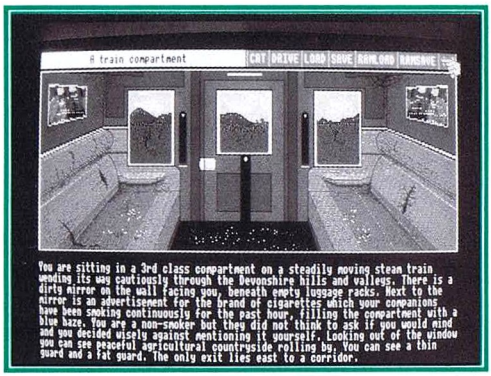
A view of each hole is drawn on screen starting from the tee with a figure preparing to take a shot. You select your club and distance and watch as the figure draws back the club and cracks the ball off down the fairway. There are the usual hazards to contend with such as trees, bunkers and water. On reaching the green, the screen displays an aerial view of the putting surface along with an indication of slope.

The graphics in *Holed Out* are excellent, although the screens do take quite a few seconds to be drawn, even on the Arc. The digitised sound effects give added realism to the game.



Holed Out

I found *Holed Out* incredibly addictive when I first played it on the Beeb, and the Arc version is even better. As a golfer myself, I think it is a first class simulation. Others who have seen the game and are not golfers also have quickly become addicted to it. In fact, I had a hard job keeping my hands on the review copy! It is fascinating to play and has lasting appeal. Again, highly recommended.



Fugitive's Quest

Finally, we have a graphic adventure called *Fugitive's Quest* (£29.95 inc. VAT) from Robico Software. The game scene is set in 1952 and you take the role of a man who has been wrongly found guilty of murdering his wife and now faces the gallows. You are being transported by train to Plymouth to face the hangman and you must escape and prove your innocence.

As usual with other Robico adventures I found the opening location a hard nut to crack but kicked myself for having been so daft when I finally succeeded in getting out of the train compartment. However, the rest of the adventure is no easy matter either! But then a good adventure should not be too easy.

It is difficult in the short space of a review to give a good insight into any adventure. Suffice to say that *Fugitive's Quest* like other Robico adventures is of a high standard and I confess to being already rather hooked on it. I would recommended it to seasoned adventurers. My only criticism is the price which at £29.95 is a bit on the steep side.

All the above games are now being stocked by BEEBUG and can be obtained with the usual discount for members.

RU

Explore the fascinating world of 3-D shapes with Archimedes **STELLATOR2**
Ray-traced shadows now 5x faster, Interrupt/restart facility, **£16.50** (inc p&p, notes, user support)
Poly Projects, 107 Appledore Avenue, Nottingham NG8 2RW

BACKUP IN BROMLEY

AUTHORISED ACORN DEALERS

We offer full technical backup, service and advice as well as a comprehensive range of software and hardware for your Archimedes. If you have a problem - we can help you.

Call in for a software or hardware demonstration, or phone for prices and availability

Data Store

6 Chatterton Road, Bromley, Kent
Telephone : 01-460 8991 (closed Wednesdays)

MEMORY ALLOCATION REVISITED

David Spencer offers some more information on configuring memory allocations under RISC OS.

Last's month RISC User included a short utility program to dynamically change the allocation of blocks of memory such as the screen RAM and the font cache, without need for a hard reset. Since then, David Pilling has pointed out that there is in fact an undocumented module to do just this on the RISC OS applications disc. However, we can be excused for not finding it earlier, as it is hidden away in the *Lander* game. To install the module, mount the *Apps 2* disc and enter:

```
QUIT
$.!Lander.Memalloc
```

Memalloc offers a total of nine star commands:

***SystemSize**	***RMASize**
***ScreenSize**	***SpriteSize**
***FontSize**	***RAMFSSize**
***RMAFree**	***SpriteFree**
***FontFree**	

The first six of these are used to set the total amount of memory allocated to the given area. The other three commands are used to set the amount of *free* space in the RMA, Sprite and Font areas independent of what is already in the particular area. All the commands have the same syntax, for example:

```
*FontSize <size> [<star command>]
```

The parameter *size* is the number of K to allocate (in decimal), and this will be rounded up according to page size. For example:

```
*ScreenSize 150
```

will in fact set the screen size to 160K.

Like last month's utility, these commands can only change memory allocations from outside an application (in other words from the star prompt). However, if the command fails in any way, it doesn't produce an error. Instead, the optional star command given after the size is executed. In the simplest case this could be made to cause an error by using the *Error command. For example:

```
*RMASize 240 Error 100 "RMASize failure"
```

Alternatively, in a more sophisticated scheme, the star command could be used to run another program which then tried to correct the source of the error. **RU**

Two Basic Compilers (continued from page 39)

practice Dabs' documentation is far more substantial and much more detailed, now providing over 230 pages of information. Yet Silicon Vision's compiler is definitely the more complete one. Their product produces faster code. It outperforms ABC2 in a number of ways - speed, easier multiple star commands for relocatable modules, and fuller array handling, for example.

Almost as important is ease of use: to the first-time ABC2 user paying nearly £100 for a compiler, there are a number of operations, particularly with the Arthur version, that are a little off-putting. On the other hand, Silicon Vision has come up with a serviceable and accessible piece of software that is easy to understand and even easier to use.

If you already have ABC1 and are wondering whether to be content with the Dabs upgrade, then it is very likely that you will be satisfied, if you can live with ABC's greater limitations. Though most programs compiled with ABC are slower than those compiled with RiscBasic, they will still be running twice as fast as the equivalent Basic on average.

If you own neither, are looking for THE compiler to go for, and are confused about which one deserves to win the very public battle that has been raging in the advertisements, well my preference is for RiscBasic, but it is a very close thing.

Product	RiscBasic
Supplier	Silicon Vision Signal House, Lyon Road, Harrow, Middx HA1 2AG.
Price	£99.95 inc. VAT and p&p.

Product	ABC2
Supplier	Dabs Press 5 Victoria Lane, Whitefield, Manchester M25 6AL.
Price	£99.95 inc. VAT and p&p.

RU

ARE YOU GOOD ENOUGH?

As the leaders in software for the Archimedes range of computers, CLARES MICRO SUPPLIES are looking to extend our range even further. We are looking for people who are as excited by the Archimedes as we are.

If you have written any programs, completed or not, then we would like to hear from you.

If you have any ideas for programs and have the ability to execute the ideas then we want to hear from you.

If you have the ability to program the Archimedes but not the ideas to program then we want to hear from you.

Programs can be written in any language as long as they perform their stated task. Many of our programs contain large chunks of BASIC with ARM code in the areas that it is needed. BASIC on the Archimedes is a very powerful language and we do not attach any snob value to its use. If your program does what is meant to do then thats all we are interested in. Why not join the top team on the Archimedes. You get the support of our in-house team, privileged access through us to Acorn and invitations to our informal programmers seminars.

The most important point is that you will be earning top royalty rates of if you prefer we will purchase your program outright.

Please write, in confidence, to Mr. D. Clare at:

Clares Micro Supplies,
98 Middlewich Road,
Northwich,
CHESHIRE CW9 7DA

If you have a program either complete or in development then please enclose a copy for our evaluation.

To protect yourself we advise that you lodge a copy of the program with your bank or solicitor BEFORE you send us a copy. You can then prove that your program pre-dates anything that we have.

Act today and become part of the leading software team producing software for the worlds fastest micro.

TECHNICAL QUERIES

COPY QUERY

Dear RISC User,

*I have had a lot of problems using *COPY. In fact I have not yet succeeded in making it work at all, though I have been able to copy files using the Desktop. Can you help?*

J.H.Waton

From the comments that you make in your letter, I see that you have RISC OS. The information on *COPY is given on pages 208-210 of the *Archimedes User Guide* supplied with the RISC OS upgrade, but it is very short on examples. So I will supply some.

To copy the file *myfile* in directory $.PROGRAMS to $.BACKUP, just use:

```
*COPY $.PROGRAMS.myfile $.BACKUP.myfile
```

One possible trap for the unwary is that you *must* supply both a source *and* destination filename. The advantage of this is that you can change the name of a file as you copy it.

If you are copying from one disc to another, you must give disc names or drive numbers, and these must be preceded by a colon. For example, to copy the file *file1* in the root directory of drive 4 to $.FILES on drive 0, with a destination file name of *filebackup*, you would use:

```
*COPY :4.$.file1 :0.$.FILES.filebackup
```

If you are using the disc name to identify the disc, the syntax is very similar. In the previous example, suppose that the name of the hard disc (drive 4) was *HardDisc*, and the name of the disc in drive 0 was *BackUp*, you could use:

```
*COPY :HardDisc.$.file1 :BackUp.$.FILES.
filebackup
```

As with all filing system commands, case is unimportant.

In the brief space available here, I have not been able to mention copying files between filing systems or the so-called *Copy Options*. The latter are detailed in the *User Guide* referred to above together with one example.

Lee Calcraft

TELEPHONE TONES

Dear RISC User,

Do you have any plans to publish a sound voice module that can generate the tones used by a tone-dial telephone? This could be used in conjunction with a telephone directory program that I am writing, in order to dial the number automatically.

R. Carter

If any member wishes to send us a sound voice module which they have written, then we will consider it for publication, as we did in Volume 1 Issue 4. However, Mr. Carter doesn't actually need a separate module for his application.

The Dual Tone Multi-Frequency (DTMF) system of telephone dialling works by transmitting two audio tones for each digit. The two tones are chosen from two groups of four, thereby giving a total of sixteen possible variations. All that is needed to dial directly from the Archimedes is to set up two sound channels using the default 'WaveSynth-Beep' voice, and then use these to generate the two tones required. The commands needed to set up the two sound channels are:

```
VOICES 2
VOICE 1,"WaveSynth-Beep"
VOICE 2,"WaveSynth-Beep"
```

The table below gives the two tones used for each.

Digit	Tone 1	Tone2
0	24063	26145
1	22288	25564
2	22288	26145
3	22288	26726
4	22979	25564
5	22979	26145
6	22979	26726
7	23482	25564
8	23482	26145
9	23482	26726

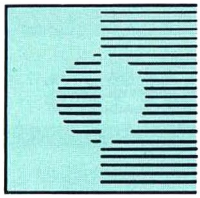
The tones can then be dialled using the following simple code:

```
SOUND 1,-15,tone1,100:SOUND 2,-15,tone2,100
t=TIME:REPEAT UNTIL TIME>t+20
SOUND 1,0,0,0:SOUND 2,0,0,0
```

This should be executed for each digit using the appropriate values for *tone1* and *tone2*. A delay loop and a second set of SOUND statements are used to cut the tones off abruptly. Obviously, the tones must be played down the telephone line, and this can normally be done by placing the mouthpiece near to the speaker. Direct connection to the phone line is neither permitted nor easy.

David Spencer

RU



Lingenuity
Specialist Software

Archimedes PRODUCTS FOR THE 1990's

NEW PRESENTER STORY

The ultimate in Business Presentation software. Turn your Archimedes into a tool to project combined images of Text, Logos, Graphs, Digitised & Video Pictures to illustrate your presentations more clearly. Many special screen features are included such as vertical and smooth scrolling and bouncing. It can be used with most makes of RGB projectors or you can use your Archimedes monitor for smaller Presentations. PRESENTER STORY is available for Business applications now. Education and Media/Video versions will be available soon. RISCOS compatible.
£199.00 + VAT

PRESENTER

The established presentation package for Archimedes users in schools and businesses alike. Colour 3-D Bar-, Line- and Pie- charts and graphs can be made quickly and easily in the Acorn WIMP environment. These charts and graphs can be printed on FX-compatible and Integrex colour printers and also on Plotmate plotter. Alternatively, they can be ported into other packages, such as 1st Word Plus, Artisan & Graphic Writer. RISCOS compatible.
£24.95 + VAT

NEW CONTROL PANEL

*Confused about the *Configure comands? With Control Panel there is no need to worry. Use Control Panel to set the right Archimedes' configuration for all your application software. You can then Save and Load the configuration settings of your choice for use at later dates. Control Panel also includes a utility to build configuration files into program boot files. Fully wimp-driven with constant real-time help screen.*
£14.95 + VAT

NEW COLOUR CONVERTER

Do you want to capture colour images onto your Archimedes screen? The Colour Converter podule running with the Watford Digitiser allows full colour images to be captured and saved as screens or sprites. The package also includes new Dithering Software which will give you greatly enhanced near-television quality pictures, using an effective palette of 256,000 colours! The Colour Converter ½ Width Podule and Software is available now.
£169.95 + VAT

NEW AIM (Archimedes Image Manager)

Public Domain Image Processing software from Delft University which will increase the contrast and sharpen blurred images that have been digitised. Many varied uses—in education, industry and the home. A multisync monitor is required in order to give the necessary resolution.
£9.00 + VAT

VISA

VISA and ACCESS accepted

All Lingenuity products are available direct or from good Acorn dealers.

Educational discounts and site-licences are available.

For further details please contact us on 098 685 477 or write to

Postbag

GETTING INTO WINDOWS

I am writing to express frustration at the apparently insuperable difficulty in understanding the WIMP system sufficiently to make use of it within programs. I have studied the series in RISC User Volume 1 Issues 5, 6 & 7, and other articles elsewhere, and of course the *Programmer's Reference Manual*. Despite considering myself computer literate, I have been beaten by the complexity of the WIMP system.

I understand the basic principles - creating a window or menu is clear enough - but opening and polling are a different story. The problem is to write programs which use the features of the system and which work. I believe that this problem must be shared by a large number of your readers. The purpose of this letter is to convince you that the problem exists, and to suggest how to solve it.

Norman Kirkby

Mr.Kirkby's letter continues by analysing what he sees as the major problems, and follows this with detailed and constructive proposals for tackling the problems highlighted above. In response, we have to say that above all else the WIMP system is complex. There is a wealth of detail that has to be mastered before even the most trivial of WIMP programs can be made to work. We have been well aware that the first series of three articles which Mr.Kirkby refers to were neither adequately detailed, nor yet sufficiently comprehensive, but at that time, with a much lower membership, the magazine was smaller and competition for space much greater.

However, as Mr.Kirkby rightly points out later in his letter, the WIMP system, particularly under RISC OS, is far too important to ignore, and we hope to rise to the challenge in future issues of RISC User. However, to do justice to the subject, and to explain it adequately will take more than a short article or two.

THE 'BOFFINS' FIGHT BACK

In the May issue of RISC User (Volume 2 Issue 6) you invite comment from 'more technically minded members'. I value RISC User for news and product reviews, but find the programs virtually useless. Programs are most useful to me for the techniques and ideas behind them, but these are rarely easy to untangle from your articles. The introductory sections are almost always too brief to provide a sensible explanation of how the program works, and the program listings are never commented. This is particularly true of assembler programs.

At this point you may cry "No room", but this is no solution. Fewer programs could be presented, but to a higher standard. Better explanation and proper commenting would give the beginner a chance of understanding the program, and would allow readers like myself to extract the bits of interest; the needs of the inexperienced and experienced are not necessarily at odds. Programs could be relegated to an appendix (lower quality paper?) or even to the magazine disc.

I hope this letter may help you to find out whether others share the same views.

Dr. Mark Wetton

As might be expected, previous comments in ***Postbag*** *have prompted several of the more technically minded readers to write in response. Clearly we shall be looking closely at the results of our reader survey, included in the previous issue, before making any changes. The question of more detailed explanations at the expense of program listings (as suggested by Dr. Wetton) was covered in the questionnaire, and is one possible change which we shall be considering quite carefully. Our thanks also to other readers who have written in with their ideas and suggestions.*

A CALL FROM DOWN UNDER

We here at the bottom of the Earth, being so far away from our sources of BBC and Archimedes information and supplies, would like to make contact with one of your clubs, but which shall it be? We would like some help in that direction as there are quite a few members on our side of the world keen to keep up to date on all the latest details.

So let's hear from you - you will at least get a reply from our members.

Theo Ollifers
Auckland BBC User Group

Of course, we assume that BBC and Archimedes owners in New Zealand are avid readers of our two magazines BEEBUG and RISC User anyway. But the personal contact between smaller and regional user groups can be very rewarding. If any group in the UK wishes to follow up the appeal from Theo Ollifers they should write to him at Auckland BBC User Group, P.O.Box 9322, New Market, Auckland, New Zealand.

RU

You ain't seen nothing yet.

PipeDream 3 - 100% RISCware

Available July 1989
See it on stand 37 at the Acorn User Show July 21-23

COLTON SOFTWARE, BROADWAY HOUSE, 149-151 ST. NEOTS ROAD, HARDWICK,
CAMBRIDGE CB3 7QJ, ENGLAND. TEL. 0954 211472 FAX. 0954 211607

HINTS & TIPS

HINTS & TIPS

This month's collection of useful Hints and Tips is rounded up by David Spencer and Lee Calcraft.

SYS STRINGS

You may be aware that a string can be passed directly to a SYS call when in fact the native SWI call expects a pointer to a string. In such cases, Basic performs all the necessary conversions. For example:

```
SYS "OS_Write0","Hello World"+CHR$0
```

It is, however, also possible to read a string back from a SYS call. Take for example:

```
DIM buffer 100
SYS "OS_SWINumberToString",&26,buffer,
100 TO ,name$
PRINT name$
```

This will convert the SWI number &26 into its corresponding name, "OS_GSRead".

The DIM serves to set up a block of memory to hold the converted name, and this is passed to the SWI call as the parameter for register R1. On exit from the call, this register still holds the pointer, and the assignment to *name$* forces Basic to read the string back into the string variable. This technique can be used in all cases where a SWI call returns a pointer to a string.

CHANGING DISCS

The RISC OS Desktop on the 300 series offers the very nice feature that if an attempt is made to access a disc which is not currently in the drive, then it will wait until the correct disc is inserted and then continue without any further action being necessary. However, the new drives used on the 400/1 series and the A3000 do not provide a Disc Changed Interrupt (DCIRQ), and this means that if the correct disc is not present, then once it has been inserted you must go and click on the 'OK' icon before the new disc is recognised.

On the good side, the new disc drives have moulded eject buttons which cannot (should not?) fly off.

APPENDING LISTING

We sometimes publish particularly long programs in RISC User in two or more parts. Examples of this are the RISC User Notepad and the Toolbox. It can be very hard to find any mistakes made when a new set of lines are added to an existing program, and a better way is to enter and check the new lines before adding them to the existing program. To do this, enter the new lines as if they formed a complete listing, check it, and save the program as a text file by using the commands:

```
*SPOOL TextProg
LIST
*SPOOL
```

Then, load the original program and add the new lines with the command:

```
*EXEC TextProg
```

The result will then be a complete program with the new lines added to the existing code.

QUICK PRINT

Under RISC OS, the call:

```
SYS "OS_PrintChar",code
```

will send to the printer the single character whose ASCII code is code. This call will work regardless of whether the printer has been enabled with VDU2, or even completely disabled using *FX 3.

INPUT/OUTPUT REDIRECTION

A major, yet little known, feature of both RISC OS and Arthur is the ability to redirect input and output during the execution of a star command. This means that all output that would normally be sent to the screen will instead go to a file, or all input that would be read from the keyboard is read instead from a file. This redirection is achieved by adding one of the following to the star command:

`{ > file }`	to send output to a file
`{ >> file }`	to append output to a file
`{ < file }`	to take input from a file

Spaces are critical, and there must be at least one space around each element, including before the '{' and after the '}'.

The redirection specification, as these are called, can be placed anywhere in the command where a space would otherwise be. For example:

```
*Basic { > Bout } -Quit MyProg
```

This will execute the Basic program MyProg, saving all the output in the file Bout, and QUITing when finished.

A frequent sight in assembly language source code is the following two lines:

```
1000 [
1010 OPT pass%
```

This has the disadvantage that the Basic assembler will print the current instruction pointer address after the '[' is encountered, even if the OPT subsequently turns off the listing. This can cause two addresses to flash on the screen while trying to assemble the code with no listing. To get around this, put the two commands on a single line, thus:

```
1000 [OPT pass%
```

TYPES DON'T MATCH ERROR

The error message "Types don't match" is given if you try to save a file (using SAVE, *SAVE or OS_File) with a name already used for a directory. If you have a large number of objects in your current directory, you might well not realise the coincidence - and this very unhelpful message only obscures the problem.

If you try to *load* a directory, you get the much more helpful message:

```
xxxx is a directory.
```

SWIS AND INTEGER VARIABLES

When using variables in SWI calls it is advisable to use integer variables, because real variables will give the error "Number too big" if values greater than &7FFFFFFF are ever assigned. This is because Basic converts the assigned value to an integer before passing it to the corresponding SWI call, and the valid range for integers is -&80000000 to +&7FFFFFFF.

LARGE PROGRAMS IN TWIN, AGAIN

In RISC User Volume 2 Issue 5 we gave a hint about overcoming Twin's 6527 line limit with Basic programs. David Pilling reports another solution. When you exit Twin with a program longer than this, Basic renumbers it so that every line is given the number "9". You can then use RENUMBER 1,1 to renumber it correctly. You can even RUN the program with all lines set to 9, providing of course, that no line number references are made.

READING THE COLOUR PALETTE

The SYS call "OS_ReadPalette" can be used to read the colour palette including the border colours. The following procedure makes use of this. It takes a single parameter: the logical colour number to be investigated. If you want information on the border, set this to -1. It returns with the variables red, green, and blue, set to the amounts of *red green* and *blue* used in the physical colour (range 0-240). For further information, see the *Programmer's Reference Manual* part 1.

```
  10 REM >Palette2
  20 MODE 12
  30 PROCreadpalette(2)
  40 PRINT"red ";red
  50 PRINT"Green ";green
  60 PRINT"Blue ";blue
  70 END
  80 :
  90 DEFPROCreadpalette(logcol)
 100 IF logcol=-1 THEN border=TRUE ELSE
 border=FALSE
 110 SYS "OS_ReadPalette",logcol,16-8*b
order TO ,,first
 120 blue=first>>>24
 130 green=(first>>>16) AND &FF
 140 red=(first>>>8) AND &FF
 150 ENDPROC
```

RISC User Magazine Disc

July/August 1989

ARCHIMEDES PROCEDURE LIBRARY

This month's procedure is a generalised input routine which allows textual data to be entered with formatting and error checking.

ARCHIMEDES VISUALS

A demonstration of the effects that can be achieved by directly reprogramming the video controller chip - VIDC.

MULTI-TASKING DISC FORMATTER

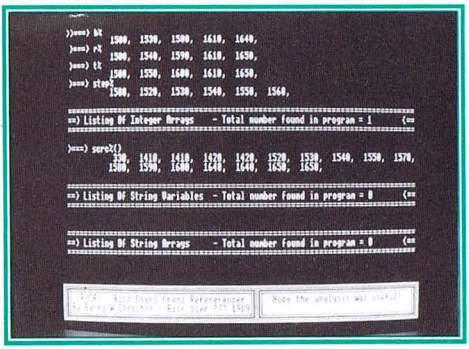
A Desktop utility which can format a floppy disc while you continue with other applications.

CROSS REFERENCER

A utility to list the names of all variables, procedures and functions used in a program, together with the numbers of the lines on which they occur.

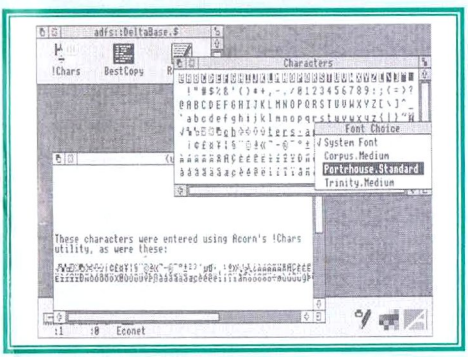
BASIC SQUEEZE

A procedure which will remove unnecessary spaces, and optionally REMs, from a Basic program.



BBC SCREEN CONVERTER This program will take a saved screen from a model B or Master, and convert it into an Archimedes format screen.

DESKTOP BASIC HANDLER A useful Desktop Utility which converts Basic programs to text files and vice versa. This enables programs to be examined and altered without leaving the Desktop.



* BONUS ITEMS *

BESTCOPY

A utility from Acorn which will help you recover files from a corrupted disc. The utility will take a named file and will try and transfer as much of it as possible to a new disc.

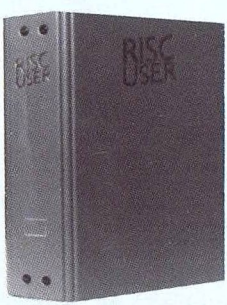
KEYBOARD UTILITY

This Desktop utility, also from Acorn, simplifies the entering of special characters, such as foreign letters. Trying to remember the necessary ALT key combination for all the characters can be fairly hard. This utility will display all the special characters on the screen, and allow them to be entered simply by clicking on the appropriate one.

ARCSCAN DATA

Index entries for this issue of RISC User and the latest BEEBUG (Vol.8 No.2) to be used with *Arcscan*.

RISC User magazine discs are available to order, or by subscription. Full details of prices etc. are given on the back cover of each issue of RISC User.



RISC User BINDERS

£3.90 + £1 p&p

BEEBUG Ltd
Dolphin Place, Holywell Hill, St. Albans,
Herts AL1 1EX
Tel. (0727) 40303

RISC USER magazine

MEMBERSHIP

RISC User is available only on subscription at the rates shown below. Full subscribers to RISC User may also take out a reduced rate subscription to BEEBUG (the magazine for the BBC micro and Master series).

All subscriptions, including overseas, should be in pounds sterling. We will also accept payment by Connect, Access and Visa, and official UK orders are welcome.

RISC USER SUBSCRIPTION RATES		RISC USER & BEEBUG
£14.50	1 year (10 issues) UK, BFPO, Ch.I	£23.00
£20.00	Rest of Europe & Eire	£33.00
£25.00	Middle East	£40.00
£27.00	Americas & Africa	£44.00
£29.00	Elsewhere	£48.00

BACK ISSUES

We intend to maintain stocks of back issues New subscribers can therefore obtain earlier copies to provide a complete set from Vol.1 Issue 1. Back issues cost £1.20 each. You should also include postage as shown:

Destination	UK, BFPO, Ch.Is	Europe plus Eire	Elsewhere
First issue	60p	£1	£2
Each subsequent issue	30p	50p	£1

MAGAZINE DISC

The programs from each issue of RISC User are available on a monthly 3.5" disc. This will be available to order, or you may take out a subscription to ensure that the disc arrives at the same time as the magazine. The first issue (with six programs and animated graphics demo) is at the special low price of £3.75. The disc for each issue contains all the programs from the magazine, together with a number of additional items by way of demonstration, all at the standard rate of £4.75.

MAGAZINE DISC PRICES	UK	Overseas
Single issue discs	£ 4.75	£ 4.75
Six months subscription	£25.50	£30.00
Twelve months subscription	£50.00	£56.00

Disc subscriptions include postage, but you should add 60p per disc for individual orders (30p for additional discs on the same order).

All orders, subscriptions and other correspondence should be addressed to:

RISC User, Dolphin Place, Holywell Hill, St Albans, Herts AL1 1EX.
Telephone: St Albans (0727) 40303
(24hrs answerphone service for payment by Connect, Access or Visa card)